ON DIAGONALLY-PRECONDITIONING THE TRUNCATED-NEWTON METHOD FOR SUPER-SCALE LINEARLY CONSTRAINED NONLINEAR PROGRAMMING

L. F. ESCUDERO

We present an algorithm for super-scale linearly constrained nonlinear programming (LCNP) based on Newton's method. In large scale programming solving the Newton equation at each iteration can be expensive and may not be justified when far from a local solution; we briefly review the current existing methodologies, such that by classifying the problems in small-scale, super-scale and supra-scale problems we suggest the methods that, based on our own computational experience, are more suitable in each case for coping with the problem of solving the Newton's equation. For super-scale problems, the Truncated-Newton method (where an inaccurate solution is computed by using the conjugate-gradient method) is recommended; a diagonal BFGS preconditioning of the gradient is used, so that the number of iterations to solve the equation is reduced. The procedure for updating that preconditioning is described for LCNP when the set of active constraints or the partition of basic, superbasic and nonbasic (structural) variables have been changed.

1. INTRODUCTION

Consider the unconstrained nonlinear programming (UNP) problem

$$\min\{F(X) \in X\ R^n\} \qquad (1.1)$$

where F(X) is a nonlinear function with the following properties: F(X) is, at least, 2-continuously differentiable; for all $X' \in R^n$ the level sets $L(X') \Delta \{X: F(X) \leq F(X')\}$ are bounded.

Let $\overset{*}{X}$ denote a weak local minimum in (1.1); that is, $\overset{*}{X}$ is a point for which $\exists \delta > 0$ such that $F(\overset{*}{X}) \leq F(X)\ \forall X: \|X - \overset{*}{X}\| \leq \delta$. (Point $\overset{*}{X}$ will be a strong local minimum if the above relation is a strict inequality for $X \neq \overset{*}{X}$). A first-order necessary condition for $\overset{*}{X}$ is that $g(\overset{*}{X}) = 0$, where $g(\overset{*}{X}) \equiv \overset{*}{g}$ is the gradient vector of F(X) evaluated at $\overset{*}{X}$. A sufficient condition for $\overset{*}{X}$ be a weak local minimum is that $\overset{*}{g} = 0$ and $G(\overset{*}{X})$ is positive definite (pd), where $G(\overset{*}{X}) \equiv \overset{*}{G}$ is the Hessian matrix of F(X) evaluated a $\overset{*}{X}$; note that a second-order necessary condition for $\overset{*}{X}$ is that $\overset{*}{G}$ be positive semi-definite.

One of the most traditional methods for obtaining $\overset{*}{X}$ is the Newton method; given an initial estimation $X^{(0)}$, it computes a sequence of stepdirections $\{d^{(k)}\}$ such that at iteration k it solves the linear system

$$G^{(k-1)} d^{(k)} = -g^{(k-1)} \qquad (1.2)$$

and iterates $X^{(k)}$ as follows

$$X^{(k)} = X^{(k-1)} + \alpha^{(k)} d^{(k)} \qquad (1.3)$$

where $\alpha^{(k)}$ is the steplength at iteration k, such that F(X) is minimized along direction $d^{(k)}$,

$$\alpha^{(k)} = \arg \min\{F(X^{(k-1)} + \alpha d^{(k)}):\ \alpha > 0\} \qquad (1.4)$$

We will refer to (1.2) as the Newton equation and $d^{(k)}$ as the Newton direction.

Problem (1.4) is termed exact linesearch; solving (1.4) is as difficult as solving (1.1); alternatively, the so-termed approximate linesearch methods are used such that for a direction descent enough, that is

$$(-g^{(k-1)} d^{(k)}) / (\|g^{(k-1)}\|_2 \|d^{(k)}\|_2) > \varepsilon \qquad (1.5)$$

where ε is a positive small tolerance that avoids the quasi-orthogonality between the gradient and the direction, the algorithm is globally convergent if $\alpha^{(k)}$ produces a strong enough reduction in F(X) and, specifically, if conditions GPW/18,31/ are satis-

- Laureano F. Escudero - Centro de Investigación UAM-IBM - Pº Castellana, 4 - Madrid-1

- Article rebut el Octubre de 1982.

fied:

(i) $|g(X^{(k-1)} + \alpha^{(k)} d^{(k)})^t d^{(k)}| \leq$

$$\leq -\eta g(X^{(k-1)})^t d^{(k)} \qquad (1.6)$$

or, alternatively /6/, if the calculation of the gradient is expensive,

$$\frac{|F(X^{(k-1)} + \alpha^{(k)} d^{(k)}) - F(X^{(k-1)} + \nu d^{(k)})|}{\alpha^{(k)} - \nu} \leq$$

$$\leq -\eta g(X^{(k-1)})^t d^{(k)} \qquad (1.7)$$

where ν is a scalar such that $0 \leq \nu < \alpha^{(k)}$, and $0 \leq \eta < 1$. Note that satisfying (1.6) for $\eta=0$, - also satisfies (1.4).

(ii) $F(X^{(k-1)}) - F(X^{(k-1)} + \alpha^{(k)} d^{(k)}) \geq$

$$\geq -\mu\, \alpha^{(k)} g(X^{(k-1)})^t d^{(k)} \qquad (1.8)$$

for $0 < \mu \leq 0.5$. Typically, $\eta=0.9$ and $\mu=10E-4$ -- such that if $\mu \leq \eta$, $\exists \alpha$ that satisfies condi--- tions GPW. Conditions (1.6) y (1.7) avoid -- $\alpha^{(k)}$ be excessively small; condition (1.8) -- avoids it be excessively large.

Newton method is important because if provides a standard with which to compare alterna te methods for solving (1.1). Its positive - and negative aspects are very well known. -- Briefly, its advantage is that the algorithm is locally and quadratically convergent; note that a sequence $\{X^{(k)}\}$ is said to converge to $\overset{*}{X}$ with a rate r if

$$0 \leq \lim \frac{||X^{(k+1)} - \overset{*}{X}||_r}{||X^{(k)} - \overset{*}{X}||_r^r} \equiv \beta < \infty \qquad (1.9)$$

The rate of convergence is linear if r=1 --- (for $0<\beta<1$) and cuadratic if r=2. The conver gence is Q-superlinear if $\beta=0$ and r=1; the - type r for which $\beta>0$ is finite defines the - order of this convergence (see e.g. /13/ for most of the concepts used in this work).

The inconveniences of the Newton method are significant; they are as follows:

(i) The method is not globally convergent -- for $\alpha^{(k)}=1$; that is, $g^{(k)} \neq 0$ for $k \to \infty$ if - $X^{(0)}$ is not close enough to $\overset{*}{X}$.

(ii) There is not solution in (1.2) if $G^{(k-1)}$ is singular.

(iii) For nonconvex problems, $G^{(k-1)}$ is not necessarily pd and, then, $d^{(k)}$ is not guaranteed to be descent.

(iv) In any case, the Hessian $G^{(k-1)}$ must - be evaluated and a n-dimensional linear system (1.2) must be solved at each ite ration k.

In sec. 2 we review the main alternate me--- thods for solving the Newton equation and motivate the Truncated-Newton method. In sec.3 we describe the algorithm model for this method. Sec. 4 describes the diagonal BFGS pre conditioning of the gradient that we suggest for the Truncated-Newton method in UNP pro-- blems. Sec. 5 briefly reviews the LCNP pro-- blem and describes the procedure for using - the Preconditioned Reduced Truncated Newton method. Sec. 6 describes the procedure for - obtaining the BFGS preconditioning in LCNP; the procedures for updating it are also described for the cases where (1) a basic varia ble changes its status, (2) a superbasic va- riable becomes nonbasic, (3) a nonbasic va-- riable becomes superbasic, (4) a nonactive - constraint becomes active, and (5) an active constraint becomes nonactive; a multiple de- activating strategy is also described. Sec.7 describes the procedure for 'solving' the -- Newton equation after the de-activating pro- cess is performed. And, finally, sec. 8 re-- ports some computational experience.

2. MOTIVATION FOR THE TRUNCATED-NEWTON METHOD

The main alternates to Newton method are as follows:

1) Quasi-Newton (QN) methods /6/.

They are the most reliable methods such - that, by calculating the pd approximation, - say $B^{(k-1)}$ of matrix $G^{(k-1)}$ and using conditions GPW, the Newton difficulties may all - be overcome with the exception of (iv); ---- their rate of convergence is superlinear. -- For non-small problems, their using is prohi bitive.

It is well known that the Broyden QN methods and, specifically, the BFGS approximation -- $B^{(k-1)}$, have been proved to be the QN me---- thods with the best performance; note that - conditions GPW guarantee $B^{(k-1)}$ to be pd and

then, $d^{(k)}$ is descent.

QN methods are used with satisfactory results in constrained nonlinear programming when the number $\bar{n}_S$ of superbasic variables (see /21/ - for the definition) is, say $\bar{n}_S \leq 300$. See /7/, /22/, /26/ among others for the nonlinear --- constrained case and /9/, /21/ for the linear ly constrained (LCNP) case.

2) Conjugate-gradient (CG) methods /11/.

This type of methods does not use the ---- Hessian $G^{(k-1)}$, nor its approximation $B^{(k-1)}$. The stepdirection $d^{(k)}$ is obtained by using the negative of the gradient $g^{(k-1)}$ and adding the 'correction' of a multiple of the previous iteration, such that the directions are conjugate if F(X) is quadratic and $\alpha^{(k)}$ is obtained with an exact linesearch; it has the - quadratic termination property, such that -- the number of iterations to obtain $\overset{*}{X}$ is $m_G \leq n$, where m_G is the number of distinct eigenva--- lues of Hessian matrix G. It may be preconditioned with a symmetric pd (spd) matrix, such that the number of distinct eigenvalues of -- the new matrix is reduced. It is based on the CG method introduced in /19/ for solving sistems of linear equations (i.e., the Newton -- equation (1.2)).

If F(X) is a general nonlinear function, a - second correction (a multiple of the so-termed restart direction) is usually added to -- the formula for $d^{(k)}$; in any case, its rate of convergence is linear, but till very re--- cently there were not other methods to cope - with supra-scale problems (say, n>600).

Conditions GPW do not guarantee $d^{(k)}$ to be -- descent for $\eta>0$; it is required to add a new condition to be satisfied while obtaining --- $\alpha^{(k-1)}$ so that additional gradient evalua---- tions are needed.

CG methods for LCNP are used in /1/, /20/, -- /21/ among others. Due to the restriction to be imposed on $\alpha^{(k-1)}$ such that step -------- $\alpha^{(k-1)} d^{(k-1)}$ is not only feasible and des--- cent but also allows $d^{(k)}$ to be descent, it happens very frecuently that conditions GPW and the aditional mentioned above are not satisfied; in this case let $\alpha^{(k-1)}$ be feasible and descent (although, not descent enough) - and reset $d^{(k)}=-g^{(k-1)}$, and hence, the information obtained from previous iterations is lost. See in /11/, /14/ critical revisions of CG-based methods when applied to LCNP problems.

3) Limited-Storage Quasi-Newton (LSQN) me--- thods /25/.

The BFGS QN stepdirection with exact line--- search may be interpreted as a CG stepdirection for which the approximation of the Hessian inverse $G^{(k-1)^{-1}}$, instead of being fi-- xed (to the identity matrix in the traditional CG) is updated at each iteration by any member of the Broyden QN methods /23/. This interpretation is of value because it motivates techniques for using limited storage to improve the CG method performance.

Based on the above remark, a new class of methods is proposed /25/, such that the Broy-- den QN updates (and, specially), the BFGS -- approximation) of matrix $G^{(k-1)^{-1}}$ use only - the last m updates of the previous itera---- tions; for m=1 the m-steps LSQN stepdirec--- tion is a CG stepdirection (if the BFGS formula is used, the previous matrix update is the identity and an exact linesearch is used) and for m=k-1 it is a QN stepdirection. Note that the QN update $H^{(k-1)}$ of $G^{(k-1)^{-1}}$ is not required to be stored, since the stepdirec-- tion $d^{(k)} = -H^{(k-1)} g^{(k-1)}$ (1.2) may be obtained from a sequence of linear combinations - of vectors that are computed by using the already available information: points $X^{(j)}$ and gradients $g^{(j)}$ for j=k-m-1,...,k-1; the need for storage depends on m and it may be drastically reduced. LSQN methods have some of - the properties of QN methods; conditions GPW alone guarantee $d^{(k)}$ to be descent. In summary, for large values of m these methods have a good rate of superlinear convergence, but the needs for storage and computation are -- not meaningless; for m=2 there is a good balance between storage and computation to be required and rate of convergence.

A 2-steps LS method has been suggested in -- /28/; it uses information from the previous and the restart iterations. It may be viewed as an extension of the CG method with restart direction that has some of the QN properties (e.g., Q-superlinear convergence); conditions GPW guarantee that $d^{(k)}$ is descent. Better -- results are obtained /14/ with a 2-steps BFGS

method, where the gradient $g^{(k-1)}$ is diagonally-BFGS preconditioned and the accumulated step $X^{(k-2)}-X^{(r)}$ is used where r is the re-start iteration.

The advantages of LSQN methods over CG methods are, basically, as follows: (a) the stepdirection is descent provided conditions GPW are satisfied and, then, additional gradient evaluations are not required in the approximate linesearch; and (b) the rate of convergence is much better, even for m=2 where the required storage is still small. Although the convergence is slower than when using QN methods, today there are not other methods to cope with supra-scale problems (say, n>600).

The 2-steps LS method introduced in /28/ for UNP has been used for LCNP in /29/. The Preconditioned 2-steps BGFS Accumulated method described in /14/ for UNP is to be used for LCNP in the sequel of this paper so that computational comparison with the Truncated-Newton approach (see below) could be provided. The additional advantage of LSQN methods over CG methods in LCNP is that, since they do not require any additional condition to conditions GPW to guarantee that the superbasic stepdirection is descent, it results that the next stepdirection $d^{(k)}$ is not to be restarted as $-g^{(k-1)}$; the reason is that there is more room for a steplength $\alpha^{(k-1)}$ that produces a feasible-descent step, being feasible since it must be $\alpha^{(k-1)} \leq \alpha_m^{(k-1)}$ (where $\alpha_m^{(k-1)}$ is an upper bound that guarantees that the bounds on variables and constraints are not violated) and descent enough since it must only satisfy conditions GPW. If there is not such a value, reset $\alpha^{(k-1)} = \alpha_m^{(k-1)}$ (such that the step is feasible and still descent since condition (1.8) is satisfied in this case) and 'correct' the LSQN matrix if it is not pd; note that conditions GPW guarantee that the matrix is pd so that the stepdirection is descent. Today, there are not other methods to be used for getting practical results when the number $\bar{n}_S$ of superbasic variables at each iteration is, say $\bar{n}_S$>600 (supra-scale problems).

4) Inexact-Newton (IN) methods /5/.

Since the benefits of the Newton and Quasi Newton directions are mainly local (i.e., in the vecinity of $\overset{*}{X}$), there appears to be no justification for expending the effort required to get an accurate solution to equation (1.2) when far from a local optimum $\overset{*}{X}$. It makes sense to give an inexact solution $d^{(k)}$ to that equation, increasing the accuracy when getting close to $\overset{*}{X}$, provided that storage and computation are reduced: a scale independent measure of this accuracy is the relative residual error

$$r^{(k)} = \|e^{(k)}\|_2 \; / \; \|g^{(k-1)}\|_2 \qquad (2.1)$$

where $e^{(k)}$ is the residual error vector when solving equation (1.2) with the given accuracy at iteration k, such that it can be written

$$e^{(k)} = B^{(k-1)} d^{(k)} + g^{(k-1)} \qquad (2.2)$$

where $B^{(k-1)}$ is the approximation of $G^{(k-1)}$.

IN methods solve Newton equation (1.2) at each major iteration k, producing a stepdirection $d^{(k)}$ such that condition $r^{(k)} \leq \eta^{(k)}$ is satisfied, where $\eta^{(k)}$ is the given tolerance in the accuracy. They have the following properties:

(i) Global convergence if $\lim \eta^{(k)}=0$ for $k\to\infty$ and $\alpha^{(k)}$ satisfies conditions GPW; the rate of convergence is superlinear depending on $\eta^{(k)}$.

(ii) Stepdirection $d^{(k)}$ is always descent, even when the equation solving is abruptly interrupted.

(iii) Then there is always solution to equation (1.2).

(iv) Hessian matrix $G^{(k-1)}$ alone is not required to be evaluated, nor approximated.

The CG method to solve systems of linear equations introduced in /19/ obtains $d^{(k)}$ by iteratively solving (1.2); at each iteration, say i , a stepdirection $d^{(i)}$ is obtained so that $\|e^{(i)}\|_2 < \|e^{(i-1)}\|_2$, where $e^{(i)}$ is defined in (2.2). A so-termed Truncated-Newton (TN) method, a IN method, 'truncates' the sequence of iterations {i} once the given accuracy is obtained. Let 'major iteration' k denote the calculation of steplength $\alpha^{(k)}$ along a given $d^{(k)}$ and the computation of $g^{(k)}$ and $B^{(k)}$. Major iterations are used to

modify the algorithm so as to control its -- global behavior; they play a significant role in early stages of the computation. Let - 'minor iteration' i at iteration k denote a given iteration for an iterative solving of Newton or Quasi-Newton equation (1.2) so --- that stepdirection $d^{(k)}$ is obtained; they -- are important when close to $\overset{*}{X}$ since, in its vecinity, $\alpha^{(k)}=1$ satisfies conditions GPW -- and the algorithm possesses the same asympto tic rate of convergence as Newton method. No te that for $\eta^{(k)}$ large enough, the TN method is the same CG method since iteration i=1 sa tisfies $r^{(i)} \leq \eta^{(k)}$ $\forall k$; for $\eta^{(k)}=0$ the TN ---- method is the same Newton (or QN) method sin ce the equation solving is not truncated. It shares the rate of convergence of the Newton methodology, but it does not require to storage the Hessian matrix $G^{(k-1)}$, nor its ---- approximation $B^{(k-1)}$; it only needs (see below) the product $G^{(k-1)}\delta^{(i)}$ or its approxima tion). Therefore, there is a direct trade-off between the amount of work required to compute a stepdirection and the accuracy with - which the Newton equation is solved. This -- method, 'intermediate' between CG and Newton methodologies, is suggested for solving su-- per-scale problems (say, $300<n\leq 600$ for UNP - and $300<\bar{n}_S\leq 600$ for LCNP).

3. TRUNCATED-NEWTON (TN) METHOD.

3.1. Skeletal algorithm.

The TN method is a natural extension of the CG method for solving system (1.2), such --- that by substituting notations $G^{(k-1)}, g^{(k-1)}$ and $d^{(k)}$ by $\bar{G}$, $\bar{g}$, and $\bar{d}$, respectively it can be written

$$\bar{G}d + \bar{g} = 0 \qquad (3.1)$$

At each minor iteration i, a stepdirection - $\delta^{(i)}$ is obtained as a linear combination of residual error $e^{(i-1)}$ and the stepdirections $\{\delta^{(j)}\}$ of previous iterations, such that --- they are conjugate (i.e., $\delta^{(i)t}\bar{G}\delta^{(j)}=0$ ---- i,j,i≠j). Let $d^{(i)}=d^{(i-1)} + \alpha^{(i)}\delta^{(i)}$ be the solution (probably, inexact) of system (3.1) where $\alpha^{(i)}$ is the steplength that solves --- (1.4) along $\delta^{(i)}$ in the UNP problem min ---- $\{e^{(i-1)t}\alpha\delta^{(i)} + 1/2\alpha^2\delta^{(i)t}\bar{G}\delta^{(i)}\}$; the residual error $e^{(i)}$ of (3.1) can be written

$$e^{(i)} = \bar{G}d^{(i)} + \bar{g} = e^{(i-1)} + \alpha^{(i)}\bar{G}\delta^{(i)} \qquad (3.2)$$

If $e^{(i)}\leq\bar{\eta}$, where $\bar{\eta}$ is the given accuracy to lerance at major iteration k, then $\bar{d}:=d^{(i)}$ - is the truncated solution of system (3.1). - The procedure is as follows.

Skeletal algorithm Al.

(0) Assign $e^{(0)}:=\bar{g}$, $\delta^{(1)}:=-e^{(0)}$, $q^{(1)}:=\bar{G}\delta^{(1)}$
If $\delta^{(1)t}q^{(1)} \leq \varepsilon_1\|\delta^{(1)}\|_2^2 \rightarrow \bar{d}:=\delta^{(1)}$

(1) $d^{(1)}:=\delta^{(1)}$, $e^{(1)}:=e^{(0)}+q^{(1)}$
Stopping rule I:
If $\|e^{(1)}\|_{1+t}/\|\bar{g}\|_{1+t}\leq\bar{\eta}_I \rightarrow \bar{d}:=d^{(1)}$, stop
i:=2

(2) $\beta^{(i)}:=\|e^{(i-1)}\|_2^2/\|e^{(i-2)}\|_2^2$
$\delta^{(i)}:=-e^{(i-1)}+\beta^{(i)}\delta^{(i-1)}$, $q^{(i)}:=\bar{G}\delta^{(i)}$
If $\delta^{(i)t}q^{(i)} \leq \varepsilon_1\|\delta^{(i)}\|_2^2 \rightarrow \bar{d}:=d^{(i-1)}$, stop

(3) $\alpha^{(i)}:=\|e^{(i-1)}\|_2^2/\delta^{(i)t}q^{(i)}$
$d^{(i)}:=d^{(i-1)}+\alpha^{(i)}\delta^{(i)}$
If $i=\tau_1 \rightarrow \bar{d}:=d^{(i)}$
$e^{(i)}=e^{(i-1)}+\alpha^{(i)}q^{(i)}$

(4) Stopping rule I:
If $\|e^{(i)}\|_{1+t}/\|\bar{g}\|_{1+t} \leq \bar{\eta} \rightarrow \bar{d}:=d^{(i)}$, stop

(5) i:=i+1, go to (2)

It can be shown /4/ that for $\varepsilon_1>0$ small ---- enough, $\bar{d}$ is a descent stepdirection (1.5), the steplength $\alpha^{(k)}=1$ satisfies conditions - GPW in the vecinity of $\overset{*}{X}$ (if $\bar{G}$ is pd and --- $\bar{\eta}_I\rightarrow 0$ for $k\rightarrow\infty$), and the above algorithm is -- globally convergent; in addition, if $\overset{*}{G}$ is pd the rate of convergence on $\{\bar{X}\}\rightarrow\overset{*}{X}$ is superlinear /5/ iff

$$\lim \|\bar{e}\| / \|\bar{g}\| \rightarrow 0 \quad (\text{i.e., } \bar{\eta}_I\rightarrow 0) \qquad (3.3)$$

such that its order is t+1, where $0<t\leq 1$ iff

$$\lim \|\bar{e}\|_{1+t} / \|\bar{g}\|_{1+t}^{1+t} < \infty \qquad (3.4)$$

Thus, tolerance $\bar{\eta}_I$ can be written

$$\bar{\eta}_I = \min\{\bar{\eta}_0, \bar{\gamma}\|\bar{g}\|_{t+1}^{t}\} \qquad (3.5)$$

for $0<\bar{\eta}_0<1$ and $\bar{\gamma}>0$; for t=1 the rate of convergence is quadratic as the Newton method. If $\|\bar{g}\|$ is large ($\bar{X}$ is away from $\overset{*}{X}$), only -- few minor iterations are required for obtain ing $\bar{d}$; when $\bar{X}$ is getting close to $\overset{*}{X}$ then --- $\|\bar{g}\|\rightarrow 0$ which implies $\bar{\eta}_I\rightarrow 0$ and, then, $\bar{d}$ is - getting close to a Newton stepdirection. Tolerance τ_1 is a safeguard against unstabilities on calculating $q^{(i)}$. Although ε_1 only - needs to be positive, it is other safeguard; it avoids that $\bar{d}$ is not descent (e.g., if $\bar{G}$

is not pd), Typical values, $\varepsilon_1=\varepsilon_M^{1/2}$, where - ε_M is the machine precision in floating point calculations (in our case, 10E-15), $\tau_1=3n$, -- $\bar{\eta}_0=1/k$ and $\bar{\gamma}=1$ /2/.

3.2 Using Hessian matrix.

Note that algorithm A1 does not require the calculation of any Hessian matrix (unlike QN methods) but the product $q^{(i)}\equiv\bar{G}\delta^{(i)}$. For super-scale problems, $\bar{G}$ is suggested to be analytically evaluated only when it is very --- sparse; if $\bar{G}$ has a special sparse structure, it could be approximated by finite differences by using the CPRT methodology /27/; note that $\bar{G}$ is available in some way for F(X) --- quadratic. Otherwise, $q^{(i)}$is approximated by finite differences such that

$$q^{(i)} \simeq (g(\bar{X}+\sigma\,\delta^{(i)}) - g(\bar{X}))/\sigma \qquad (3.6)$$

Note that an additional gradient evaluation is required at each minor iteration; it does not add any great inconvenience if the problem is sparse with many constant or zero -- gradient elements; note that CG methods require, at least, one gradient evaluation per each related iteration. For $\sigma\to 0$ the rate of convergence is still superlinear if the gradient is analytically evaluated; otherwise, the approximation (3.6) may be unstable /15/. Typical value,

$$\sigma=\varepsilon_M^{1/2}/\|\delta^{(i)}\|_2 .$$

4. PRECONDITIONING THE TRUNCATED NEWTON (PTN) METHOD.

We have stated that the number of iterations to obtain $\bar{X}^*$ is $m_G\leq n$ if F(X) is quadratic, G is pd, the directions are conjugate and the steplength satisfies (1.4). So, the number - of iterations will be reduced if the original problem is substituted by other equivalent problem where $m_R<m_G$, being R the new -- Hessian matrix.

Let W be a symetric pd (spd) matrix. The solution $\bar{d}$ to system (3.1): $\bar{G}\bar{d}=-\bar{g}$ (and, then, the solution to the related unconstrained --- quadratic problem) can be obtained by solving system

$$W^{-1/2}\bar{G}W^{-1/2}d_W = -W^{-1/2}\bar{g} \qquad (4.1)$$

such that $\bar{d}=W^{-1/2}\bar{d}_W$. Let $\bar{R}\equiv W^{-1/2}\bar{G}W^{-1/2}$ and, - then, $W^{-1/2}\bar{R}W^{1/2}=W^{-1}\bar{G}$. It can be shown /15/ that $\bar{R}$ has the same eigenvalues as $W^{-1}\bar{G}$. So, a target could be to precondition $\bar{G}$ with a matrix W^{-1}, such that matrix $W^{-1}\bar{G}$ has a --- great number of eigenvalues close to unity. Since the condition number κ of a matrix -- can be expressed as $|\lambda_{max}|/|\lambda_{min}|$, where -- λ_{max} and λ_{min} are the largest and smallest eigenvalues, respectively, it results that smaller κ,greater probability of more --- (normalized) eigenvalues close to unity.

Once a suitable matrix W is identified, algorithm A1 can be used to obtain the solution (probably, inexact) $\bar{d}_W$ to system ----- $\bar{R}d_W=-\bar{g}_W$, where $\bar{g}_W=W^{-1/2}\bar{g}$; $\delta_W^{(i)}$, $d_W^{(i)}$ and $e_W^{(i)}=\bar{R}d_W^{(i)}+\bar{g}_W$ are the stepdirection solution, estimate and residual error at minor iteration i, respectively. Calculation of matrix $W^{-1/2}$ is unstable and, so are the calculations of $\bar{g}_W$ and $\bar{R}$; but, matrix $W^{-1/2}$ is not required to be used since equivalences ----- $\bar{g}_W\equiv W^{-1/2}\bar{g}$, $\delta_W^{(i)}\equiv W^{1/2}\delta^{(i)}$, $d_W^{(i)}\equiv W^{1/2}d^{(i)}$ $q_W^{(i)}\equiv W^{-1/2}q^{(i)}$ and $e_W^{(i)}\equiv W^{-1/2}e^{(i)}$ may be --- used, such that

$$W^{1/2}d^{(1)} = -W^{-1/2}\bar{g} \;\therefore\; d^{(1)} = -W^{-1}\bar{g}$$
$$W^{-1/2}q^{(1)} = -W^{-1/2}\bar{G}\bar{g} \;\therefore\; q^{(1)} = -\bar{G}\bar{g}$$

and

$$W^{1/2}\delta^{(i)} = -W^{-1/2}e^{(i-1)}+\beta^{(i)}W^{1/2}\delta^{(i-1)} \;\therefore\; \delta^{(i)} =$$
$$= -W^{-1}e^{(i-1)}+\beta^{(i)}\delta^{(i-1)}$$
$$W^{-1/2}q^{(i)} = W^{-1/2}\bar{G}\delta^{(i)} \;\therefore\; q^{(i)} = G\delta^{(i)}$$

where

$$\beta^{(i)} = \|W^{-1/2}e^{(i-1)}\|_2^2/\|W^{-1/2}e^{(i-2)}\|_2^2 =$$
$$= e^{(i-1)t}W^{-1}e^{(i-1)}/e^{(i-2)t}W^{-1}e^{(i-2)}$$

such that

$$W^{1/2}d^{(i)} = W^{1/2}d^{(i-1)}+\alpha^{(i)}W^{1/2}\delta^{(i)} \;\therefore\; d^{(i)} =$$
$$= d^{(i-1)}+\alpha^{(i)}\delta^{(i)}$$

where

$$\alpha^{(i)} = \frac{e^{(i-1)t}W^{-1}e^{(i-1)}}{\delta^{(i)t}W^{1/2}W^{-1/2}q^{(i)}}$$

and, finally

$$W^{-1/2}e^{(i)} =$$
$$= W^{-1/2}e^{(i-1)}+\alpha^{(i)}W^{-1/2}q^{(i)} \;\therefore\; e^{(i)} = e^{(i-1)}+\alpha^{(i)}q^{(i)}$$

Of course, stepdirection $\delta^{(i)}$ and solution - estimate $d^{(i)}$ obtained as above have different values to those obtained, by using algorithm A1, in the original problem.

Note that in the above approach W^{-1} ia not - required, but $W^{-1}\bar{g}$ and $W^{-1}e^{(i-1)}$; computatio nally, obtaining W^{-1} is more unstable than - obtaining $W^{-1}b$, where b is a vector. Since - W is spd, it can be expressed as W=LU, where L is a lower triangular matrix with identity diagonal and U is an upper triangular matrix such that vector x that solves Wx=b can be obtained with the following stable procedure: obtain $\underline{x}$ in $L\underline{x}=b$ and obtain x in $Ux=\underline{x}$. The - LU factorization of symmetric non-pd matrix is unstable. In summary, the preconditioned algorithm A1 is as follows.

Skeletal algorithm A2

(0) Assign $e^{(0)} := \bar{g}$
Obtain $z^{(0)}$: $Wz^{(0)} = e^{(0)}$
$\delta^{(1)} := -z^{(0)}$, $q^{(1)} := \bar{G}\delta^{(1)}$
If $\delta^{(1)t} q^{(1)} \leq \varepsilon_1 \|\delta^{(1)}\|_2^2 \rightarrow \bar{d} := \delta^{(1)}$, stop
(1) $d^{(1)} := \delta^{(1)}$, $e^{(1)} := e^{(0)} + q^{(1)}$
Stopping rule I:
If $\|e^{(1)}\|_2 / \|\bar{g}\|_2 \leq \bar{\eta}_I \rightarrow \bar{d} := d^{(1)}$, stop
i := 2
(2) Obtain $z^{(i-1)}$: $Wz^{(i-1)} = e^{(i-1)}$
$\beta^{(i)} := e^{(i-1)t} z^{(i-1)} / e^{(i-2)t} z^{(i-2)}$
$\delta^{(i)} := -z^{(i-1)} + \beta^{(i)} \delta^{(i-1)}$, $q^{(i)} := \bar{G}\delta^{(i)}$
If $\delta^{(i)t} q^{(i)} \leq \varepsilon_1 \|\delta^{(i)}\|_2^2 \rightarrow \bar{d} := d^{(i-1)}$, stop
(3) $\alpha^{(i)} := e^{(i-1)t} z^{(i-1)} / \delta^{(i)t} q^{(i)}$
$d^{(i)} := d^{(i-1)} + \alpha^{(i)} \delta^{(i)}$
If $i = \tau_1 \rightarrow \bar{d} := d^{(i)}$, stop
$e^{(i)} := e^{(i-1)} + \alpha^{(i)} q^{(i)}$
(4) Stopping rule I:
If $\|e^{(i)}\|_2 / \|\bar{g}\|_2 \leq \bar{\eta}_I \rightarrow \bar{d} := d^{(i)}$, stop
(5) i := i+1, go to (2)

The values of tolerances ε_1, τ_1 and $\bar{\eta}_I$ are - as in algorithm A1.

Obtaining matrix W.

Matrix W can be identified in several ways /24/. Basically, they obtain the LSQN approximation of matrix G^{-1}; it is fixed after a given number of updatings /23/ or it is upda ted al each major iteration /2/. In the se-- cond case, $W^{-1} \equiv H^{(k-1)}$ has been obtained by using the last, say m previous major itera-- tions; note that the matrix is spd. If $\alpha^{(j)}$ - has been obtained with an exact linesearch - for j=k-m,...,k-1 and F(X) is quadratic, the QN condition is satisfied; that is,

$$H^{(k-1)} y^{(j)} = p^{(j)} \qquad j=k-m,\ldots,k-1 \qquad (4.2)$$

where $y^{(j)} \equiv g^{(j)} - g^{(j-1)}$ and $p^{(j)} \equiv \alpha^{(j)} d^{(j)}$; - since F(X) is quadratic,

$$Gp^{(j)} = y^{(j)} \qquad j=1,\ldots,n \qquad (4.3)$$

and, then

$$H^{(k-1)} Gd^{(j)} = d^{(j)} \qquad j=k-m,\ldots,k-1 \qquad (4.4)$$

Note that matrix $H^{(k-1)}G$ has, at least, m eigenvalues equal to unity; in that case, - $m_R \leq n-m+1$. On the other hand, LSQN update -- $B^{(k-1)}$ of G is more stable than update ---- $H^{(k-1)}$ of G^{-1} and $B^{(k-1)-1}$ is not necessari- ly $H^{(k-1)}$.

Using a QN update for preconditioning G (and then, $g^{(k-1)}$) has some theoretical justifica tion, since it can be shown /23/ that, under the same hypothesis for which (4.2) is satis fied, the CG method preconditioned with a spd matrix W (algorithm A2 for $\bar{\eta}_I=0$) and, - then, W may be the identity matrix (algorithm A1 for $\bar{\eta}_I=0$) and the BFGS (QN) method produce identical stepdirection if in the latter ---- $H^{(0)} := W^{-1}$. The same results may be obtained - for $\bar{\eta}_I>0$ and, then, for the TN method.

Now, since the CG method is an alternate to - the QN method such that the need for storage is reduced, instead of calculating $H^{(k-1)}$, -- the product $H^{(k-1)}g^{(k-1)}$ is directly obtained; then, there is not any matrix alone to be --- used. This product may be obtained with the - sequence of linear combinations

$$H^{(k-m)}g^{(k-1)} = \text{lc}\{H^{(k-m-1)}g^{(k-1)}, H^{(k-m-1)}y^{(k-m)}, p^{(k-m)}\}$$

$$H^{(k-m+1)}g^{(k-1)} = \text{lc}\{H^{(k-m)}g^{(k-1)}, H^{(k-m)}y^{(k-m+1)}, p^{(k-m+1)}\}$$

.....

$$H^{(k-1)}g^{(k-1)} = \text{lc}\{H^{(k-2)}g^{(k-1)}, H^{(k-2)}y^{(k-1)}, p^{(k-1)}\}$$

where $H^{(k-m-1)}=I$; note that $H^{(j-1)}y^{(j)}$ for j=k-1,...,k-m+1 is obtained following the sa me procedure used for $H^{(j-1)}g^{(k-1)}$. The pro- cedure does not require a great amount of -- storage for reasonable values of m (say, --- $2 \leq m \leq 4$), although it is time consuming; it is more unstable than using $B^{(k-1)}$, but it is - prohibitive with Limited Storage.

The preconditioning can be directly applied for F(X) being a general nonlinear function

/25/; note that, in this case, matrix $H^{(k-1)}$ satisfies (4.2) only for m=1.

A more simple preconditioning consists in -- assigning $W:=\bar{B}^{(k-1)}$, where $\bar{B}^{(k-1)}$ is the diagonal of the BFGS matrix /6/ $B^{(k-1)}$, such -- that $B^{(0)}=I$ and

$$\bar{B}_\ell^{(k)} = \bar{B}_\ell^{(k-1)} + \frac{1}{y^{(k)t} p^{(k)}} y_\ell^{(k)2} + \frac{1}{g^{(k-1)t} d^{(k)}} g_\ell^{(k-1)2} \qquad (4.5)$$

for $\ell=1,\ldots,n$ and $k>1$. Note that $\bar{B}_\ell$ is the -(ℓ,ℓ)-th element of matrix B. The new matrix W requires small storage; it is very stable since:

(a) If $y^{(k-1)t} p^{(k-1)} \leq \varepsilon_2$ or $\exists \bar{B}_\ell^{(k-1)} \leq \varepsilon_2$, $\bar{B}^{(k-1)}$ is not guarantee to be pd; in this case, the matrix is not updated. ε_2 is a small positive tolerance; typically, ε_2=10E-4. Note that the previous update is always pd.

(b) If the condition number κ of $\bar{B}^{(k-1)}$ is -large (e.g., $\kappa>\kappa_m$ where $\kappa_m=1/(100n^{1/2}\varepsilon_M)$ -- and

$$\kappa = \max_{(\ell)} \bar{B}_\ell^{(k-1)} / \min_{(\ell)} \bar{B}_\ell^{(k-1)} \qquad (4.6)$$

$\bar{B}_\ell^{(k-1)}$ is substituted /15/ by $\bar{B}_\ell^{(k-1)^w}$ $\forall \ell$, where $w=\log \kappa_m / \log \kappa$.

5. LINEARLY CONSTRAINED NONLINEAR PROGRAMMING

5.1. Introduction.

The linearly constrained nonlinear programming (LCNP) problem is

$$\text{minimize } \{F(X) \; X \in F \subseteq R^n\} \qquad (5.1)$$

where

$$F \triangleq \{X \mid \bar{b} \geq AX \geq \underline{b}, \; U \geq X \geq l\} \qquad (5.2)$$

where A is an m.n matrix, m<n, and F(X) is a general nonlinear twice continuously diffe-- rentiable function, at least, for feasible - points such that for all $\bar{X} \in F$ the level sets

$$L(\bar{X}) \triangleq \{X \in F, \; F(X) \leq F(\bar{X})\} \qquad (5.3)$$

are bounded. Let M be the set of constraints, E be the set of equality constraints (such - that $i \in E$ if $\bar{b}_i=\underline{b}_i$), and J be the set of variables. Let $\overset{*}{A}$ be the $\overset{*}{t}$.n matrix of active --- constraints at a local optimal point, say $\overset{*}{X}$ and $\overset{*}{b}$ the $\overset{*}{t}$-vector of right-hand-side corresponding to $\overset{*}{A}$ (i.e., $\overset{*}{A}\overset{*}{X}=\overset{*}{b}$), such that $\overset{*}{t}=|\overset{*}{W}|$ where $\overset{*}{W}$ is the set of active constraints -- and $\overset{*}{b}_i=\bar{b}_i \vee \underline{b}_i$ for $i \in \overset{*}{W}$. Let $\overset{*}{V}$ be the set of - active variables at $\overset{*}{X}$, such that $j \in \overset{*}{V}$ if --- $\overset{*}{X}_j=U_j \vee l_j$ and $\overset{*}{r}=|\overset{*}{V}|$. Let $\overset{*}{I}$ be the r,n matrix of active bounds at $\overset{*}{X}$, such that it is the n.n identity matrix I from where the row related to variable $j \notin \overset{*}{V}$ has been deleted.

Note that $AX-Y=\underline{b}$, $\bar{b}-\underline{b} \geq Y \geq 0 \leftrightarrow \bar{b} \geq AX \geq \underline{b}$; then $i \in \overset{*}{W}$ for $\overset{*}{Y}_i=0 \vee \bar{b}_i$. The X-variables are termed --- structural; the Y-variables are termed --- slack.

Because the constraints are a linear system, the properties of linear subspaces make it possible to state a simple characterization of all feasible moves from a feasible point. Consider the move between two feasible ----- points $\overset{*}{X}$ and $\bar{X}$ along the manifold defined by the sets $\overset{*}{W}$ and $\overset{*}{V}$; by linearity $\overset{*}{A}(\overset{*}{X}-\bar{X})=0$ and $\overset{*}{I}(\overset{*}{X}-\bar{X})=0$ since $\overset{*}{A}\overset{*}{X}=\overset{*}{b}$, $\overset{*}{A}\bar{X}=\overset{*}{b}$ and $\overset{*}{X}_j=\bar{X}_j$ $\forall j \in \overset{*}{V}$ -- and,then,

$$\overset{*}{A}d = 0, \quad \overset{*}{I}d = 0 \qquad (5.4)$$

where d is the stepdirection from $\overset{*}{X}$ to $\bar{X}$ --- such that $\bar{X}=\overset{*}{X}+\alpha d$. Any vector d for which --- (5.4) holds is a feasible stepdirection from $\overset{*}{X}$ with respect to the above manifold; it -- is also an active stepdirection; it will be descent if $F(\bar{X})<F(\overset{*}{X})$. Steplength is required to be $0<\alpha \leq \alpha_m$. where $_m$ defines the maximum allowed steplength such that $\bar{X}$ is still feasible.

Let us define a non-active stepdirection d as the feasible stepdirection such that --- some constraint or bound is removed from the sets $\overset{*}{W}$ and $\overset{*}{V}$, respectively; a feasible stepdirection d is non-active if $\exists i \in M-E \cap \overset{*}{W}$ for -- which $A_i d>0$ if $\overset{*}{Y}_i=0$, $A_i d<0$ if $\overset{*}{Y}_i=\bar{b}_i-\underline{b}_i$, or - $\exists j \in \overset{*}{V}$ for which $d_j>0$ if $\overset{*}{X}_j=l_j$, $d_j<0$ if $\overset{*}{X}_j=U_j$.

5.2. Optimality conditions.

The necessary optimality conditions for $\overset{*}{X}$ -- being a weak local minimum are as follows -- /12/, /15/.

(i) $\overset{*}{X} \in F$ (feasible).

(ii) The reduced gradient vector, say $\overset{*}{h}$ of F(X) vanishes, such that

$$\overset{*}{h} \equiv \overset{*}{Z}^t \overset{*}{g} = 0$$

where $\overset{*}{Z}$ is a $n.(n-\overset{*}{t}-\overset{*}{r})$ full column rank matrix, whose columns form the null basis of the range of matrix $(\overset{*}{A}^t, \overset{*}{I}^t)^t$ and, then,

$$\overset{*}{A}\overset{*}{Z}=0, \quad \overset{*}{I}\overset{*}{Z}=0 \tag{5.6}$$

Based on (5.4) and (5.5), we may note that any linear combinations of the columns of $\overset{*}{Z}$ give an active stepdirection d,

$$d=\overset{*}{Z}d_S \tag{5.7}$$

where d_S is a $(n-\overset{*}{t}-\overset{*}{r})$-vector termed reduced stepdirection (or superbasic stepdirection) such that a vector d that cannot be expressed by (5.7) is not an active stepdirection in the manifold defined by $\overset{*}{W}$ an $\overset{*}{V}$.

Any point at which the reduced gradient h - vanishes (5.5) is termed constrained stationary point.

The result (5.5) implies that $\overset{*}{g}$ must be a - linear combination of the rows of $\overset{*}{A}$ and $\overset{*}{I}$,

$$\overset{*}{g} = \overset{*}{A}^t\overset{*}{\mu} + \overset{*}{I}^t\overset{*}{\lambda} \tag{5.8}$$

for some vectors $\overset{*}{\mu}$ and $\overset{*}{\lambda}$; they are termed - the Lagrange multipliers of the active --- constraints and bounds, respectively; vice-versa, (5.8) implies (5.5).

(iii) Uniqueness of the Lagrange multipliers.

Let us partition matrix $\overset{*}{A}$ and gradient $\overset{*}{g}$ such that $\overset{*}{A}=(\overset{*}{B}\overset{*}{S},\overset{*}{N})$ and $\overset{*}{g}=(\overset{*}{g}_{BS}^t,\overset{*}{g}_N^t)^t$, where $\overset{*}{N}$ is a $\overset{*}{t}.\overset{*}{r}$ matrix defined by sets $\overset{*}{W}$ and $\overset{*}{V}$, - and $\overset{*}{g}_N$ is the gradient of set $\overset{*}{V}$. Based in -- (5.8), $\overset{*}{\lambda}$ can be written,

$$\overset{*}{\lambda} = \overset{*}{g}_N - \overset{*}{N}^t\overset{*}{\mu} \tag{5.9}$$

such that $\overset{*}{\mu}$ satisfies the linear system

$$\overset{*}{g}_{BS} = (\overset{*}{B}\overset{*}{S})^t\overset{*}{\mu} \tag{5.10}$$

Point $\overset{*}{X}$ does not require A to be a full row rank matrix, but the uniqueness of vectors $\overset{*}{\mu}$ and $\overset{*}{\lambda}$ requires $\overset{*}{B}\overset{*}{S}$ to have that property. -

In any case, computational stability in the algorithms that obtain the sequence $\{X^{(k)}\}\to\overset{*}{X}$ requires $\{A_i\}$ to be linearly independent for $i\in M$.

(iv) The sign of the Lagrange multipliers -- must be as follows:

$\overset{*}{\mu}_i \gtrless 0$ for $i\in E$ (equality constraint).

$\overset{*}{\mu}_i=0$ for $i\notin\overset{*}{W}$ (non-active inequality cons--- traint).

$\overset{*}{\mu}_i\geq 0$ for $i\in M-E\cap\overset{*}{W}$ such that $\overset{*}{Y}_i=0$ (active inequality constraint whose associa-- ted slack variable has the value - zero).

$\overset{*}{\mu}_i\leq 0$ for $i\in M-E\cap\overset{*}{W}$ such that $\overset{*}{Y}_i=\bar{b}_i-\underline{b}_i$ (active inequality constraint whose asso-- ciated slack variable takes its -- upper bound).

$\overset{*}{\lambda}_j=0$ for $j\notin\overset{*}{V}$ (non-active variable).

$\overset{*}{\lambda}_j\geq 0$ for $j\in\overset{*}{V}$ such that $\overset{*}{X}_j=l_j$ (active variable at its lower bound).

$\overset{*}{\lambda}_j\leq 0$ for $j\in\overset{*}{V}$ such that $\overset{*}{X}_j=U_j$ (active variable at its upper bound).

The set $D_1\cup D_2$, where $D_1\triangleq\{i\in M-E\cap\overset{*}{W}$ for $\overset{*}{\mu}_i=0\}$ and $D_2\triangleq\{j\in\overset{*}{V}$ for $\overset{*}{\lambda}_j=0\}$ is termed degenerate set of active constraints and bounds.

(v) Positive semi-definiteness of the ----- Hessian matrix.

The reduced Hessian matrix $\overset{*}{H}$ must be positive semi-definite, where

$$\overset{*}{H} \equiv \overset{*}{Z}^t\overset{*}{G}\overset{*}{Z} \tag{5.11}$$

If the degenerate set $D_1\cup D_2$ is not empty -- then the positive semi-definiteness property of $\overset{*}{G}$ must be extended to the non-active step direction d for which it holds

$A_i d>0\wedge\overset{*}{Y}_i=0\wedge i\in D_1$, or $A_i d<0\wedge\overset{*}{Y}_i=\bar{b}_i-\underline{b}_i\wedge i\in D_1$, or

$d_j>0\wedge\overset{*}{X}_j=l_j\wedge j\in D_2$, or $d_j<0\wedge\overset{*}{X}_j=U_j\wedge j\in D_2$.

Conditions (i)-(ii) and (iv)-(v) are necessary conditions for local optimality; if --- Hessian matrix is required in (v) to be positive definite then they are sufficient conditions.

5.3. Skeletal algorithm.

Following a traditional approach /21/, let - the active constraints matrix, say $\bar{A}$ be partitioned as

$$\bar{A}d = (\bar{B},\bar{S},\bar{N})\begin{pmatrix} d_B \\ d_S \\ d_N \end{pmatrix} = 0 \tag{5.12}$$

where the basic stepdirection d_B is used to satisfy the constraints set, the superbasic stepdirection d_S is allowed to vary to minimize F(X) (5.1) and the nonbasic stepdirection d_N is zero, such that set $\bar{V}$ is fixed at any of their bounds. Here $\bar{B}\bar{S} \equiv (\bar{B},\bar{S})$ and $\bar{B}$ is a $\bar{t}.\bar{t}$ non-singular matrix. At each iteration, the problem then becomes determining vector $d=(d_B^t, d_S^t, d_N^t)^t$ so that it is feasible descent. Let $\bar{P}$, $\bar{Q}$ and $\bar{V}$ denote the sets of structural basic, superbasic and nonbasic - variables, respectively; and $\bar{P}_s$ and $\bar{V}_s$ the sets of slack basic and nonbasic variables, respectively. Let $\bar{s} \equiv |\bar{P}_s|$, $\bar{v} \equiv |\bar{V}_s|$ and $\bar{n}_S \equiv |\bar{Q}|$ where $\bar{n}_S = n - \bar{t} - \bar{r}$; recall $\bar{t} \equiv |\bar{P}|$ and $\bar{r} \equiv |\bar{V}|$. -- Since $d_N = 0$ and d_S is allowed to be free, it results

$$d_B = -\bar{B}^{-1}\bar{S}d_S \qquad (5.13)$$

such that the variable-reduction characterization of matrix Z can be written

$$Z = \begin{pmatrix} -B^{-1}S \\ I \\ 0 \end{pmatrix} \qquad (5.14)$$

so that (5.7) holds.

The unconstrained reduced problem of minimizing F(X) in the manifold defined by $\bar{W}$ and $\bar{V}$ can be expressed as a function of the current superbasic set of variables; its quadratic - approximation can be written

$$\text{minimize} \quad \{h^t d_S + 1/2 d_S^t \bar{H} d_S\} \qquad (5.15)$$

where $\bar{h}$ and $\bar{H}$ are given by (5.5) and (5.11), respectively. Note that $\bar{h}$ can also be ------ written

$$\bar{h} = \bar{g}_S - \bar{S}^t\bar{\mu}_B \qquad (5.16)$$

where $\bar{\mu}_B$ solves de linear system

$$\bar{g}_B = \bar{B}^t\mu_B \qquad (5.17)$$

Note that $\bar{g}_{BS} \equiv (\bar{g}_B^t, \bar{g}_S^t)^t$ where $\bar{g}_B$ and $\bar{g}_S$ are the basic and superbasic gradients, respectively. Theoretically the algorithm continues till $||\bar{h}|| = 0$ or the superbasic set is empty and,- then, the de-activating process is executed; the Lagrange multipliers if the solution is 'optimal' or their estimates if the solution is 'quasi-optimal' are used for selecting -- the nonbasic variable to be de-activated.

The active constraints Lagrange multipliers estimates may be obtained by solving system (5.17), basic estimation or by minimizing the residual error on solving system (5.10), basic-superbasic estimation by using the QR factorization /8/; see in /11/, /12/ an ex-- tensive discussion on the subject.The Lagrange multipliers estimates of the structural - nonbasic variables are obtained by using formula (5.9).

In the de-activating process, a (structural or slack) nonbasic variable with favorable - Lagrange multiplier estimate $\bar{\lambda}_j$ $j \epsilon \bar{V}$ or $\bar{\mu}_i$ $i \epsilon M - E \cap \bar{W}$ will be chosen to be de-activated. - If not, the optimum point $\bar{X}$ has been found; otherwise, the selected nonbasic variable becomes superbasic and, if it is a slack variable, it is exchanged with an appropiate ---- basic variable such that, in this case, a -- new reduced unconstrained UNP problem is to be solved.

While minimizing in a given manifold, it is possible that either a basic or a superbasic variable strikes a bound during the search.- If a superbasic variable strikes a bound --- then it becomes nonbasic, the cardinality of the basic-superbasic set (the manifold) is - reduced by one, and the search continues. If a basic variable strikes a bound then it is exchanged with an appropiate superbasic va-- riable, and the resulting new superbasic variable is made nonbasic. See the details of the whole process in /9/. It is out of the - scope of this work to describe the different treatment to be given to slack and structu-- ral variables, structural unbounded and ---- bounded variables, and structural bounded - pure linear, linear with variable coeffi---- cient and nonlinear variables. In any case, note that slack variables will only be superbasic while changing the status from basic - to nonbasic and viceversa. A structural va-- riable is defined to be pure linear if its - coefficient in the objective function is --- constant; it is linear with variable-coefficient if, for a given value of the other variables that are used in the same objective function terms, these are a linear function of the given variable; it is nonlinear if, - at least, there is an objective function -- term that is a nonlinear function of the given variable if the rest of the variables -- are fixed. An example is as follows: ------

$F(X)=4X_1+X_2\log X_3$; variable X_1 is pure linear, variable X_2 is linear with variable-coeffi-- cient, and variable X_3 is nonlinear. The --- rough procedure for major iteration $k\geq 1$ is - as follows.

Skeletal algorithm A3.

Let $\bar{H}\equiv H^{(k-1)}$ and $\bar{h}\equiv h^{(k-1)}$. Let $\bar{A}=(\bar{B},\bar{S},\bar{N})$ --- also include the submatrix related to slack variables (i.e., sets $\bar{P}_s$ and $\bar{V}_s$), so that $\bar{B}$ - is related to sets $\bar{P}$ and $\bar{P}_s$ (being an $(\bar{t}+\bar{s})$. $.(\bar{t}+\bar{s})$ nonsingular matrix) and $\bar{N}$ is related to sets $\bar{V}$ and $\bar{V}_s$ (being an $(\bar{t}+\bar{s}).(\bar{r}+\bar{v})$ ma--- trix).

(1) Analyse stopping criteria on optimizing the current reduced unconstrained UNP - problem on the given manifold. If they are - satisfied then a 'quasi-optimal' or 'optimal' solution to this problem is found, go to (6).

(2) Obtain a (descent) superbasic stepdirec- tion $\bar{d}_S\equiv d_S^{(k)}$ on the current UNP and, by using (5.13), obtain a feasible basic step-- direction $\bar{d}_B$. Note that $\bar{d}_S$ is obtained by -- using problem (5.15) and, then, by solving - linear system

$$\bar{H}d_S = -\bar{h} \qquad (5.18)$$

In our case, the method to be used is termed Preconditioned Reduced Truncated Newton ---- (PRTN) method.

(3) Obtain the steplength $\bar{\alpha}\equiv\alpha^{(k)}$ such that, by using an approximate linesearch, con- ditions GPW are satisfied in problem

$$\min\{F(\bar{X}+\alpha\bar{d}):\ 0<\alpha\leq\bar{\alpha}_m\} \qquad (5.19)$$

where $\bar{X}\equiv X^{(k-1)}$ is the feasible solution ob-- tained at the previous iteration, and $\bar{\alpha}_m\equiv\alpha_m^{(k)}$ is the maximum allowed value, at the current iteration, for α such that $\bar{\alpha}\bar{d}$ is still feasi- ble. If there is not any value $0<\alpha\leq\bar{\alpha}_m$ that - satisfies conditions GPW, reset $\bar{\alpha}:=\bar{\alpha}_m$; note that $\bar{\alpha}\bar{d}$ is still feasible-descent.

(4) In any case, a new point $X^{(k)}=\bar{X}+\bar{\alpha}\bar{d}$ is -- obtained and its gradient $(g_B^{(k)t}, g_S^{(k)t})^t$ is evaluated or approximated. In our case, - the diagonal BFGS update $\bar{B}_S^{(k)}$ (4.5) related - to the current reduced UNP is also obtained as a function of $\bar{B}_S^{(k-1)}$, $\bar{p}\equiv\bar{\alpha}\bar{d}_S$, $\bar{y}\equiv h^{(k)}-\bar{h}$ -- and $h^{(k)}$, where $h^{(k)}$ is the reduced gradient (5.16) related to current iteration k; see - sec. 6. Reset $\bar{X}:=X^{(k)}$, $\bar{g}:=g^{(k)}$, $\bar{h}:=h^{(k)}$.

(5) If $\bar{\alpha}=\bar{\alpha}_m$, identify the active (structural or slack) variable and change its basic- superbasic status so that the basic, superba sic and nonbasic sets are updated and, so, - matrix $\bar{A}$. In our case, matrix $\bar{B}_S\equiv\bar{B}_S^{(k)}$ is also updated; see sec. 6. Go to (2).

If $\bar{\alpha}<\alpha_m$, go to (1).

(6) De-activating strategy. Let D be the -- set of (structural and slack) nonbasic - variables with favorable Lagrange multiplier estimate to be made inactive. If $D=\emptyset$, stop - since point $\overset{*}{X}$ is assumed to be found. Note that it is possible $|D|>1$; it is the case -- for which a multiple de-activating strategy is allowed such that as many as possible non basic variables are de-activated up a given bound, say $\tau_2(\bar{r}+\bar{v})$. Although more computatio nal experience is required, it seems that -- allowing $|D|>1$ has better performance /3/, - /17/, /29/ than the single de-activating -- strategy, provided that τ_2 is not too-large; typically, $\tau_2=0.05$. Our computational results (see sec.8) are also better for $|D|>1$.

(7) If $|D|>0$, update $\bar{A}$, $\bar{h}$ and, in our case, update $\bar{B}_S$ and obtain the new superbasic stepdirection d_S; see sec. 7. Obtain the new basic stepdirection d_B and go to (3).

5.4. The PRTN method. Skeletal algorithm A4.

Linear system (5.18) is solved by using the TN methodology with basically the same algo rithm A2 where $\bar{g}$, $\bar{G}$, $\delta^{(i)}$ and $\bar{d}$ are substi- tuted by $\bar{h}$, $\bar{H}$, $\delta_S^{(i)}$ and $\bar{d}_S$, respectively. - Tolerances ε_1 and τ_1 are as follows: ------- $\varepsilon_1=\varepsilon_M^{1/2}$ and $\tau_1=3\bar{n}_S$. Tolerance $\bar{n}_I$ of stopp-- ing rule I, step (4) is obtained by using -- formula (3.5), where $\bar{g}$ is substituted by $\bar{h}$, $\bar{\gamma}=1$ and $\bar{n}_0=1/k'$ where $k'\geq 1$ is the major ite ration of the subproblem defined by current sets $\bar{W}$ and $\bar{V}$; k' is reset to 1 when superba- sic set $\bar{Q}$ is changed or the active bounds on sets $\bar{V}$ and $\bar{V}_s$ have been changed. Precondi--- tion W in algorithm A2 is, now, expressed by diagonal matrix $\bar{B}_S^{(k-1)}$; see sec. 6.

Recently /17/, the following rule has been -

suggested to stop the execution of algorithm A4 in step (4).

Stopping rule II

$$|\Phi^{(i-1)}-\Phi^{(i)}|/|\Phi^{(i)}|\leq\eta_{II} \qquad (5.20)$$

where $\Phi^{(i)}$ is the value at minor iteration i of the related quadrattic function (5.18). - Note that $\Phi^{(0)}=0$ and

$$\Phi^{(1)}=\bar{h}^t d_S^{(1)}+1/2 d_S^{(1)t}\bar{H}d_S^{(1)}$$
$$=-e^{(0)t}z^{(0)}+1/2\delta_S^{(1)t}\bar{H}\delta_S^{(1)}\equiv-e^{(0)t}z^{(0)}+$$
$$+1/2\delta_S^{(1)t}q^{(1)} \qquad (5.21)$$

where all elements are already available. -- For $i\geq 2$, $\Phi^{(i)}$ can be expressed by using ---- (3.2) as follows.

$$\Phi^{(i)}=\bar{h}^t d_S^{(i)}+1/2\, d_S^{(i)t}\bar{H}d_S^{(i)}$$
$$=\Phi^{(i-1)}+e^{(i-1)t}\alpha^{(i)}\delta_S^{(i)}+1/2\,\alpha^{(i)2}\delta_S^{(i)t}\bar{H}\delta_S^{(i)} \qquad (5.22)$$

such that $\delta_S^{(i)t}\bar{H}\delta_S^{(i)}\equiv\delta_S^{(i)t}q^{(i)}$ is already available; then it results that only one -- inner product is required in its computation. One of the possible ways to overcome this inconvenience is as follows. Note that ------- $e^{(i-1)t}\delta_S^{(i)}$ can be expressed by using (3.2) as follows.

$$e^{(i-1)t}\delta_S^{(i)}=$$
$$=e^{(i-1)t}(-z^{(i-1)}+\beta^{(i)}\delta_S^{(i-1)})$$
$$=-e^{(i-1)t}z^{(i-1)}+\beta^{(i)}e^{(i-1)t}\delta_S^{(i-1)}$$
$$=-e^{(i-1)t}z^{(i-1)}+\beta^{(i)}(e^{(i-2)}+\alpha^{(i-1)}q^{(i-1)})^t\delta_S^{(i-1)}$$
$$=-e^{(i-1)t}z^{(i-1)}+\beta^{(i)}(e^{(i-2)t}z^{(i-2)}+e^{(i-2)t}\delta_S^{(i-1)})$$
$$=-e^{(i-1)t}z^{(i-1)}+e^{(i-1)t}z^{(i-1)}+\beta^{(i)}e^{(i-2)t}\delta_S^{(i-1)}$$
$$=\beta^{(i)}e^{(i-2)t}\delta_S^{(i-1)} \qquad (5.23)$$

where also all elements are already available. Hence, $\Phi^{(i)}$ can be computed very easily by using (5.22) and (5.23). Note that for i=2 the inner product $e^{(i-2)t}\delta_S^{(i-1)}$ in (5.23) -- can be expressed

$$e^{(0)t}\delta_S^{(1)}=-e^{(0)t}z^{(0)} \qquad (5.24)$$

In summary, stopping rule II (5.20), where - the convergence tolerance η_{II} must be non-negative (typically, $\eta_{II}=0.1$) is not time-consuming and does not require special storage; we have not used it yet, but the results reported in /17/ are very promising.

The preconditioned algorithm, where matrix - W is expressed by the diagonal BFGS matrix $\bar{B}_S\equiv\bar{B}_S^{(k-1)}$ related to point $\bar{X}\equiv X^{(k-1)}$, is as follows.

Skeletal algorithm A4

Set tolerances

$\varepsilon_1=\varepsilon_M^{1/2}$, $\bar{\eta}_I=\min\{1/k', \|\bar{h}\|_2\}$

$\eta_{II}=0.1, \tau_\Phi=10E\text{-}12, \tau_1=3\bar{n}_S$

(0) Assign $e^{(0)}:=\bar{h}$

$z^{(0)}:=e^{(0)}/\bar{B}_S$, $\delta_S^{(1)}:=-z^{(0)}$, $q^{(i)}:=\bar{H}\delta_S^{(1)}$

If $\delta_S^{(1)t}q^{(1)}\leq\varepsilon_1\|\delta_S^{(1)}\|_2^2\rightarrow\bar{d}_S:=\delta_S^{(1)}$, stop

(1) $d_S^{(1)}:=\delta_S^{(1)}$, $e^{(1)}:=e^{(0)}+q^{(1)}$

Stopping rule I:

If $\|e^{(1)}\|_2/\|\bar{h}\|_2\leq\bar{\eta}_I\rightarrow\bar{d}_S:=d_S^{(1)}$, stop

Assign $e^{(0)t}\delta_S^{(1)}:=-e^{(0)t}z^{(0)}$

$\Phi^{(1)}:=e^{(0)t}\delta_S^{(1)}+0.5\delta_S^{(1)t}q^{(1)}$

$i:=2$

(2) Assign $z^{(i-1)}:=e^{(i-1)}/\bar{B}_S$

$\beta^{(i)}:=e^{(i-1)t}z^{(i-1)}/e^{(i-2)t}z^{(i-2)}$

$\delta_S^{(i)}:=-z^{(i-1)}+\beta^{(i)}\delta_S^{(i-1)}$, $q^{(i)}:=\bar{H}\delta_S^{(i)}$

If $\delta_S^{(i)t}q^{(i)}\leq\varepsilon_1\|\delta_S^{(i)}\|_2^2\rightarrow\bar{d}_S:=d_S^{(i-1)}$

(3) $\alpha^{(i)}:=e^{(i-1)t}z^{(i-1)}/\delta^{(i)t}q^{(i)}$

$d_S^{(i)}:=d_S^{(i-1)}+\alpha^{(i)}\delta_S^{(i)}$

If $i=\tau_1\rightarrow\bar{d}_S:=d_S^{(i)}$, stop

Assign $e^{(i-1)t}\delta_S^{(i)}:=\beta^{(i)}e^{(i-2)t}\delta_S^{(i-1)}$

$e^{(i)}:=e^{(i-1)}+\alpha^{(i)}q^{(i)}$

(4) Stopping rule I:

If $\|e^{(i)}\|_2/\|\bar{h}\|_2\leq\bar{\eta}_I\rightarrow\bar{d}_S:=d_S^{(i)}$, stop

$\Phi^{(i)}=\Phi^{(i-1)}+\alpha^{(i)}e^{(i-1)t}\delta_S^{(i)}+0.5\alpha^{(i)2}\delta_S^{(i)t}q^{(i)}$

Stopping rule II:

If $|\Phi^{(i)}|>\tau_\Phi\wedge|\Phi^{(i-1)}-\Phi^{(i)}|/|\Phi^{(i)}|\leq\eta_{II}\rightarrow\bar{d}_S:=d_S^{(i)}$, stop

(5) $i:=i+1$, go to (2)

The aditional amount of storage required in the above algorithm is as follows: 5 $\bar{n}_S$-vectors for $e^{(i-1)}$, $z^{(i-1)}$ (although it may not be necessary if $e^{(i-1)}/\bar{B}_S$ is used), $q^{(i)}$, -- $\delta_S^{(i-1)}$ and $d_S^{(i)}$, and 15 scalars for ------ $\|e^{(i)}\|_2$, $e^{(i-1)t}z^{(i-1)}$, $e^{(i-1)t}\delta_S^{(i)}$, $e^{(i-2)t}z^{(i-2)}$, $\|\delta_S^{(i)}\|_2^2$, $\alpha^{(i)}$, $\beta^{(i)}$, $\|\bar{h}\|_2$, $\delta_S^{(i)t}q^{(i)}$, $\Phi^{(i)}$, $\Phi^{(i-1)}$, $\bar{\eta}_I$, η_{II}, τ_Φ and τ_1; only 3 inner products ($\|e^{(i)}\|_2$, $\|\delta_S^{(i)}\|_2^2$ and $e^{(i-1)t}z^{(i-1)}$) are performed at each iteration besides the

operations required for $q^{(i)}$ (see below). Note that $e^{(0)}$ and $z^{(0)}$ are not needed provided that in step (0): $\delta_S^{(1)} := -\bar{h}/\bar{B}_S$, $e^{(0)t} z^{(0)} := -\bar{h}^t \delta_S^{(1)}$, and in step (1): $e^{(1)} := \bar{h} + q^{(1)}$, $e^{(0)t} \delta_S^{(1)} := -e^{(0)t} z^{(0)}$

5.5. Using Reduced Hessian matrix. Skeletal algorithm A5.

Note that algoritm A4 does not require the calculation of any Hessian matrix, but the product $q^{(i)} \equiv \bar{H} \delta_S^{(i)}$. Note also that $\bar{H}$ (5.11) is usually a dense matrix, even if $\bar{G}$ and A are sparse matrices. In any of the possible ways to obtain $q^{(i)}$ such that

$$q^{(i)} = \bar{Z}^t \bar{G} \bar{Z} \delta_S^{(i)} \qquad (5.25)$$

the intermediate vector $\delta^{(i)}$,

$$\delta^{(i)} = \bar{Z} \delta_S^{(i)} \qquad (5.26)$$

must be obtained; matrix $\bar{Z}$ (5.14) is not explicitly calculated. By assuming that factors L and U of basic matrix $\bar{B}$ are updated, the procedure for obtaining (5.26) is as follows. (Note that suffix ℓ in the vectors to be used refers to the element related to variable ℓ in the given vector and it does not necessarily refer to the position of the element in the vector; this remark also holds for the matrices and vectors to be used in the rest of the work).

(1) Assign $\delta_\ell^{(i)} := 0 \ \forall \ell \in \bar{V} \cup \bar{V}_S$, $\delta_\ell^{(i)} := \delta_{S_\ell}^{(i)} \ \forall \ell \in \bar{Q}$
(2) $\underline{\delta}_S^{(i)} := -\bar{S} \delta_S^{(i)}$
(3) Obtain $\delta_B^{(i)}$: $LU \delta_B^{(i)} = \underline{\delta}_S^{(i)}$
(4) $\delta_\ell^{(i)} := \delta_{B_\ell}^{(i)} \ \forall \ell \in \bar{P} \cup \bar{P}_s$

Once $\delta^{(i)}$ is obtained, $\bar{G}\delta^{(i)}$ must be calculated. For super-scale problems, $\bar{G}$ is suggested to be analytically evaluated only if it is very sparse; if $\bar{G}$ has a special sparse structure, it could be approximated by finite differences by using the CPRT methodology /27/; note that $\bar{G}$ is available in some way for F(X) quadratic. Otherwise, vector $\bar{G}\delta^{(i)}$ is approximated by finite differences such that

(5)
$$\underline{\bar{\delta}}^{(i)} := (g(X^{(k-1)} + \sigma\delta^{(i)}) - g(X^{(k-1)}))/\sigma \approx \bar{G}\delta^{(i)}$$

where only the basic-superbasic elements of gradient $g(X^{(k-1)} + \sigma\delta^{(i)})$ are evaluated (or approximated). As usual, $\sigma = \epsilon_M^{1/2} / \|\delta^{(i)}\|_2$.

Finally, the procedure for obtaining $q^{(i)} \equiv \bar{Z}^t \underline{\bar{\delta}}^{(i)}$ is similar to the procedure for obtaining $\bar{h} = \bar{Z}^t \bar{g} = \bar{g}_S - \bar{S}^t \bar{B}^{t-1} \bar{g}_B$; see (5.16) and (5.17). Note that $\bar{g}_B \equiv g_B^{(k-1)}$ and $\bar{g}_S \equiv g_S^{(k-1)}$ are the gradients of sets $\bar{P} \cup \bar{P}_S$ and $\bar{Q}$, respectively evaluated at point $\bar{X} \equiv X^{(k-1)}$ such that

$$(6) \quad q^{(i)} := \underline{\bar{\delta}}_S^{(i)} - \bar{S}^t \bar{B}^{t-1} \underline{\bar{\delta}}_B^{(i)} \equiv \underline{\bar{\delta}}_S^{(i)} - \bar{S}^t \hat{\mu}$$

Vector $\hat{\mu} \equiv \bar{B}^{t-1} \underline{\bar{\delta}}_B^{(i)}$ is obtained in $(LU)^t \mu = \underline{\bar{\delta}}_B^{(i)}$ as vector $\bar{\mu}_B$ (LP simplex multipliers) is obtained in (5.17). Note that $\underline{\bar{\delta}}_B^{(i)}$ and $\underline{\bar{\delta}}_S^{(i)}$ are the subvectors of $\underline{\bar{\delta}}^{(i)}$ related to sets $\bar{P} \cup \bar{P}_S$ and $\bar{Q}$, respectively.

We may see that obtaining $q^{(i)}$ only requires at each minor iteration i, one evaluation of the basic-superbasic gradient and the solving of two linear systems with the basic matrix (whose LU factorization is available). Note that the gradient elements related to pure linear variables are constant; the elements related to linear variables whose variable coeficient is only related to nonbasic variables do not change from $\bar{X}$ to $\bar{X} + \sigma\delta^{(i)}$. Note also that the solution of the two linear systems consists in multiplying (premultiplying in one case and postmultiplying in the other) the inverse of the (nonsingular) basic matrix by a given vector. And, finally, note that $\bar{g}_{B_\ell} = 0$ for $\ell \in P_s$ and, then, $\underline{\bar{\delta}}_\ell^{(i)} = 0$ in step (5) for $\ell \in \bar{P}_s$ such that $\hat{\mu}_i = 0$ being $i \notin \bar{W}$ (see secs. 5.1 and 5.2) the nonactive inequality constraint related to slack variable ℓ; note also that slack variables cannot be used as superbasic for obtaining $\bar{d}_S$.

6. PRECONDITIONING IN LCNP.

Recall that the preconditioning matrix in algoritm A4 is expressed by the diagonal BFGS matrix $\bar{B}_S \equiv \bar{B}_S^{(k-1)}$ related to point $\bar{X} \equiv X^{(k-1)}$.

6.1. Obtaining $\bar{B}_S^{(k)}$ in current reduced problem UNP. Skeletal algorithm A6.

(0) At iteration k=0, assign $\bar{B}_{S_\ell}^{(0)} := 1 \quad \forall \ell \in \bar{Q}$. We do not recommend to obtain $\bar{Z}^t \bar{G} \bar{Z}$ even for F(X) quadratic and $\bar{G}$ sparse. Before obtaining $\bar{B}_S^{(1)}$ (but after obtaining $X^{(1)}$, matrix $\bar{B}_S^{(0)}$ is scaled /30/ so that the new ma-

trix $\bar{B}_S^{(0)}$ can be written

$$\bar{B}_{S_\ell}^{(0)} = \bar{y}^t\bar{p}/\|\bar{p}\|_2^2 \quad \ell\in Q$$

where $\bar{y}\equiv h^{(1)}-h^{(0)}$ and $\bar{p}\equiv X^{(1)}-X^{(0)}$.

At iteration $k\geq 1$ matrix $\bar{B}^{(k)}$ is calculated - (after $X^{(k)}$ is obtained) by using the BFGS - formula /6/ for the approximation of matrix $\bar{Z}^t\bar{G}\bar{Z}$ such that

(1) $\bar{B}_S^{(k)} := \bar{B}_S^{(k-1)}$, if $\bar{y}^t\bar{p} \leq \varepsilon_2$

$$\bar{B}_{S_\ell}^{(k)} := \bar{B}_{S_\ell}^{(k-1)} + \frac{1}{\bar{y}^t\bar{p}}\bar{y}_\ell^2 + \frac{1}{\bar{h}^t d_S^{(k)}}\bar{h}_\ell^2 \quad \forall \ell\in\bar{Q},$$

otherwise note $\bar{y}\equiv h^{(k)}-\bar{h}$, $\bar{p}\equiv\alpha^{(k)}d^{(k)}$ and $\bar{h}\equiv h^{(k-1)}$, and ε_2=10E-4.

(2) If $\exists\bar{B}_{S_\ell}^{(k)} \leq \varepsilon_2 \rightarrow \bar{B}_{S_\ell}^{(k)} := \bar{B}_{S_\ell}^{(k-1)}$

(3) If $\kappa>\kappa_m \rightarrow \bar{B}_{S_\ell}^{(k)} := \bar{B}_{S_\ell}^{(k)^w} \quad \forall \ell\in\bar{Q}$

where $\kappa_m = 1/(100\bar{n}_S^{1/2}\varepsilon_M)$, $\kappa = \max_{(\ell)}\bar{B}_{S_\ell}^{(k)} / \min_{(\ell)}\bar{B}_{S_\ell}^{(k)}$ and $w=\log\kappa_m/\log\kappa$.

6.2. Updating $\bar{B}_S^{(k)}$ when a structural basic - variable strikes a bound.

Let us assume that structural basic variable, say $p\in\bar{P}$ strikes a bound and, then, a pivot-- ing with an appropiate superbasic variable,- say $q\in\bar{Q}$ is performed such that, at the end - of the process, $\bar{P}\leftarrow\bar{P}\cup\{q\}/\{p\}$ and $\bar{Q}\leftarrow\bar{Q}\cup\{p\}/\{q\}$. The related updating of vector $\bar{B}_S\equiv\bar{B}_S^{(k)}$ is as follows.

Since $\bar{B}=LU$ for the old set $\bar{P}$, it results --- from (5.14) that

$$\bar{z} \equiv -(U^{-1}L^{-1})_p\bar{S} \qquad (6.1)$$

where $\bar{z}$ is the $\bar{n}_S$-row vector related to va-- riable p in old matrix $\bar{Z}$; let $\bar{z}_q$ be the pi-- vot element and v the $\bar{n}_S$-row vector such --- that

$$v = (-\bar{z}+e_q^t)/\bar{z}_q \equiv (z_1,\ldots,-1+z_q,\ldots,z_{\bar{n}_S}) \qquad (6.2)$$

where e_q is the unit $\bar{n}_S$-column vector with 1 as the q-th element.

Note that after pivoting in (p,q) $q\in\bar{P}$ and $p\in\bar{Q}$; then, the new matrix has the following expression

$$Z = \bar{Z}(I+e_q v) \qquad (6.3)$$

where I is the $\bar{n}_S.\bar{n}_S$ identity matrix and $\bar{Z}$ - is given in (5.14). Note that column vector Z_q related to variable 'p' is positioned in matrix Z as vector $\bar{Z}_q$ in matrix $\bar{Z}$.

By assuming that, say C is such that $\bar{B}_S\equiv\bar{Z}^tC\bar{Z}$ is the diagonal BFGS approximation of the re duced Hessian matrix $\bar{H}\equiv\bar{Z}^t\bar{G}\bar{Z}$, it results that its update can be written

$$\bar{B}_S := Z^tCZ =$$
$$= (\bar{Z}(I+e_q v))^t C\bar{Z}(I+e_q v) = (I+e_q v)^t\bar{Z}^tC\bar{Z}(I+e_q v)$$
$$= (I+e_q v)^t\bar{B}_S(I+e_q v) \qquad (6.4)$$

The rough procedure is as follows.

Skeletal algorithm A7

(1) Obtain row vector $\bar{z}$: $U^tL^tz = e_p$

(2) Assign $\bar{z} := \bar{z}\bar{S}$

$$z_q := -1/\bar{z}_q,\ z_\ell := \bar{z}_\ell z_q \ \forall\ell\in\bar{Q}/\{q\}$$

(3) Assign $\bar{B}_{S_\ell} := \bar{B}_{S_\ell} + \bar{B}_{S_q} z_\ell^2 \ \forall\ell\in\bar{Q}/\{q\}$, $\bar{B}_{S_q} := \bar{B}_{S_q} z_q^2$

Since z and $\bar{z}$ are not required to be saved,- only one temporary additional $\bar{n}_S$-vector is - needed in algorithm A7.

6.3. Updating $\bar{B}_S^{(k)}$ when a superbasic variable strikes a bound.

Let us assume that variable, say $q\in\bar{Q}$ strikes a bound; q is a superbasic variable in the - current iteration, or it is a (structural or slack) basic variable and an exchange with - an appropiate superbasic variable has been already performed. In this case, $\bar{Q}\leftarrow\bar{Q}/\{q\}$, $\bar{n}_S\leftarrow\bar{n}_S-1$ and, so $\bar{B}_{S_q}$ and $\bar{h}_q$ are deleted from $\bar{B}_S$ and $\bar{h}$, respectively such that $\bar{V}_s\leftarrow\bar{V}_s\cup\{q\}$ and $\bar{v}\leftarrow\bar{v}+1$ if q is slack, and $\bar{V}\leftarrow\bar{V}\cup\{q\}$and ---- $\bar{r}\leftarrow\bar{r}+1$ if q is structural. Note that $\bar{B}_{S_q}$ is - deleted from $\bar{B}_S$ only after $\bar{B}_S$ has been upda- ted and, then, $\bar{y}_q$ and $\bar{p}_q$ have been used.

6.4. Updating $\bar{B}_S^{(k)}$ when a structural nonbasic variable is de-activated.

Let us assume that structural nonbasic varia ble, say $q\in\bar{V}$ has been selected to become non active in the next iteration and, so it is - made superbasic. In this case, $\bar{V}\leftarrow\bar{V}/\{q\}$, ---- $\bar{Q}\leftarrow\bar{Q}\cup\{q\}$, $\bar{r}\leftarrow\bar{r}-1$, $\bar{n}_S\leftarrow\bar{n}_S+1$ and $\bar{h}_q=\bar{\lambda}_q$, where - this element is related to the new superbasic variable q.

The new reduced Hessian matrix will be ----- $H\equiv Z^t\bar{G}Z$, where Z (5.14) can be expressed

$$Z = (\bar{Z}\ \underline{z}) = \begin{pmatrix} \underline{z} \\ \bar{Z}\ \underline{e}_{\bar{n}_S+1} \\ 0 \end{pmatrix} \qquad (6.5)$$

where the column vector $\underline{z}$ is the solution of system

$$LUz = -\bar{N}_q \qquad (6.6)$$

such that

$$H = Z^t \bar{G} Z = \begin{pmatrix} \bar{Z}^t \bar{G} \bar{Z} & \bar{Z}^t \bar{G} \underline{z} \\ \underline{z}^t \bar{G} \bar{Z} & \underline{z}^t \bar{G} \underline{z} \end{pmatrix} \qquad (6.7)$$

Note that the old and new diagonal BFGS matrices are intended to approximate $\bar{H}$ and H, respectively. For $|D|=1$ (only one variable to be de-activated), it makes sense to express the new element $\bar{B}_{S_q}$ in $\bar{B}_S$ related to variable q as follows

$$\bar{B}_{S_q} = \begin{cases} \underline{z}^t \bar{G} \underline{z}, & \text{if } \underline{z}^t \bar{G} \underline{z} > \varepsilon_2 \\ 1, & \text{otherwise} \end{cases} \qquad (6.8)$$

Vector $\underline{z}$ is obtained with a similar procedure to step (3) of algorithm A5 (see sec. 5.5) and $\bar{G}\underline{z}$ is obtained with its step (5) where $\delta^{(i)}$ is substituted by $\underline{z}$. Recall tolerance $\varepsilon_2>0$ avoids possible unstabilities. The remarks made in that step are also applied here; note that q is the unique superbasic variable to be used and slack variables have no elements in objective function F(X); note also that the gradient elements of basic variables and variable q that are linear with variable-coefficient only related to superbasic and nonbasic variables will not change.- Since there is a direct trade-off between the amount of work required to compute a stepdirection and the accuracy with which the Hessian matrix is preconditioned, it is suggested to set $\bar{B}_{S_q}=1$, instead of computing $\underline{z}^t \bar{G} \underline{z}$ if the basic set $\bar{P}$ and variable q are not nonlinear.

If $|D|>1$ it is suggested to set the new elements of $\bar{B}_S$ to 1, since the computation of the first part of (6.8) may be too-much time consuming without any guarantee that they will not be reset to 1.

6.5. Updating $\bar{B}_S^{(k)}$ when a nonactive constraint is activated.

Let us assume that slack basic variable, say $p \in \bar{P}_s$ strikes a bound; it is the case where the related nonactive constraint, say $i \notin \bar{W}$ becomes active. An appropiate superbasic variable, say $q \in \bar{Q}$ is selected such that the pivoting (p,q) is performed and algorithm A7 (see sec. 6.2) is used for updating $\bar{B}_S$ such that $\bar{P}_s \leftarrow \bar{P}_s/\{p\}$, $\bar{Q} \leftarrow \bar{Q} \cup \{q\}$, $\bar{P} \leftarrow \bar{P} \cup \{q\}$, $\bar{s} \leftarrow \bar{s}-1$ and $\bar{t} \leftarrow \bar{t}+1$. Since p is a slack variable the procedure suggested in sec. 6.3 must be followed such that $\bar{Q} \leftarrow \bar{Q}/\{p\}$, $\bar{V}_s \leftarrow \bar{V}_s \cup \{p\}$, $\bar{n}_S \leftarrow \bar{n}_S - 1$ and $\bar{v} \leftarrow \bar{v}+1$.

6.6. Updating $\bar{B}_S^{(k)}$ when an active constraint is de-activated.

Let us assume that slack nonbasic variable, say $q \in \bar{V}_s$ has been selected to become nonactive in the next iteration; it is the case where the related active (inequality) constraint, say $i \in M-E \cap \bar{W}$ becomes nonactive. The procedure is as follows:

(a) Use the procedure suggested in sec. 6.4 such that $\bar{V}_s \leftarrow \bar{V}_s/\{q\}$, $\bar{Q} \leftarrow \bar{Q} \cup \{q\}$, $\bar{v} \leftarrow \bar{v}-1$ and $\bar{n}_S \leftarrow \bar{n}_S+1$. Note that $N_q = -e_q$ (see (5.12) and sec. 6.1) and, then, the solution $\underline{z}$ of system (6.6) is the column vector related to (inequality) constraint i in inverse basic matrix $\bar{B}^{-1}$. Note also that the superbasic variable with nonzero element in vector $\underline{z}$ is a slack variable.

(b) Select an appropiate structural basic variable, say $p \leftarrow \bar{P}$ such that the pivoting (p,q) is performed and algorithm A7 (see sec. 6.2) is used for updating $\bar{B}_S$ so that $\bar{P}_s \leftarrow \bar{P}_s \cup \{q\}$, $\bar{P} \leftarrow \bar{P}/\{p\}$, $\bar{Q} \leftarrow \bar{Q} \cup \{p\}/\{q\}$, $\bar{s} \leftarrow \bar{s}+1$ and $\bar{t} \leftarrow \bar{t}-1$.

7. 'SOLVING' THE NEWTON EQUATION ONCE THE DE-ACTIVATING PROCESS HAS BEEN PERFORMED. SKELETAL ALGORITHM A8

Let $d_S = (\underline{d}_S^t, \underline{d}_{\bar{S}}^t)^t$ be the new superbasic step direction, where $\underline{d}_S$ is the direction related to the old superbasic set $\bar{Q}$ and $\underline{d}_{\bar{S}}$ is related to set D. The following stepdirection d_S is suggested /3/ such that, although the Newton equation $Hd=-h$ will not be solved with full accuracy, it is always descent and is not time-consuming.

$$\underline{d}_S = \begin{cases} 0 \text{ if } \bar{Q}=\emptyset \vee \|\bar{h}\|_2 \leq \varepsilon_3 \\ \text{PRTN direction in } \bar{H}d_S=-\bar{h} \end{cases} \qquad (7.1)$$

$$\underline{d}_{S_q} = -h_q \qquad \forall q \in D \qquad (7.2)$$

where $\underline{d}_S$, $\bar{Q}$ and $\bar{h}$ are related to the old superbasic set (such that $\bar{h}$ is the reduced gradient related to the 'optimal' or 'quasi-optimal' solution $\bar{X}+\bar{\alpha}\bar{d}_S$), $h_q \equiv \bar{\lambda}_q$ and $\varepsilon_3 > 0$ is a dynamically adjusted tolerance that is being used in the criteria for defining when an 'optimal' solution has been reached in the minimization of the manifold defined by the old sets $\bar{V}$ and $\bar{V}_s$.

Note that if $||\bar{h}||_2 > \varepsilon_3$ it is not too-large since $\bar{X}+\bar{\alpha}\bar{d}$ is, at least, a 'quasi-optimal' solution and, then, $\bar{\eta}_I$ is relatively small (see sec. 5.4). We may note that d_S (7.1)-(7.2) is a mixture of the stepeest direction and a relatively accurate Newton direction.

8. SOME COMPUTATIONAL RESULTS:

This section reports the results of running the diagonal BFGS PRTN method in two real-life problems; see table 1. Both problems involve the management of hydroelectric systems with multiple reservoirs. The goal is to determine the amount of water to be released from and storage in each reservoir in each period (week) of a planning horizon (a year), so that the weekly power demand is satisfied at 'minimum cost'; the objective function includes a convex, nondecreasing and piecewise differenciable function of the cost of energy defecit (or surplus) determined by the energy market structure, and a nonseparable function of the reward for final reservoir storage. Problem I is well-conditioned; Problem II is very ill-conditioned. The gradients are analytically evaluated.

The PRTN method is included in the LCNP algorithm described in /9/. The main strategies are as follows.

(1) Slack variables are either basic or nonbasic such that, once a nonbasic slack variable has been selected to be de-activated, it is made basic; the reason is that the range $\bar{b}-\underline{b}$ is large enough and, hopefully, the related inequality constraint will have lesser chance of striking a bound than 'critical' structural variables. For the same reason, basic structural variables have priority over slack variables for being made superbasic.

(2) When the single de-activating strategy is used, nonbasic structural variables have priority over slack variables for being de-activated.

(3) The partitions of structural variables into pure linear, linear with variable-coefficient and nonlinear, and basic, superbasic and nonbasic variables are used for gradient evaluations and solving LP basic-superbasic and LP basic-nonbasic problems at each major iteration whenever it is possible /9/.

(4) Basic nonlinear variables have some priority over linear variables for being made superbasic, and superbasic linear variables have some priority over nonlinear variables for being made basic. The goal is to keep the basic set as close as possible to a linear set, provided that stability in the basic matrix is preserved.

(5) Partial and multiple pricing and partial exact normalization strategies /9/ are not used in this experiment; they are being used in the computational comparison to be reported in the sequel of this paper between the PRTN method and the diagonally preconditioned 2-steps BFGS with accumulated step method /14/.

(6) Problem I is optimized by, alternatively, using the PRTN method and the RQN method; the latter uses the BFGS Quasi-Newton update of the reduced Hessian matrix and solves the Newton equation with full accuracy. The RQN method uses the basic-superbasic estimation $\bar{\lambda}_{BS}$ /9/,/10/ of the Lagrange multipliers together with the basic estimation $\bar{\lambda}_B$; the PRTN method only uses the basic estimation. Both methods use the zero or near-to-zero strategy described in /12/ for de-activating a nonbasic variable.

(7) The RQN method uses the approach described in /9/ for obtaining the new superbasic stepdirection once a nonbasic variable is de-activated.

(8) The RQN method usually uses the single de-activating strategy; the PRTN method also uses the multiple strategy with the upper bounds $\min\{5, 0.01(\bar{v}+\bar{r})\}$ and $\min\{20, 0.05(\bar{v}+\bar{r})\}$.

(9) Both methods obtain the first partition into basic, superbasic and nonbasic variables by using the procedure described in /9/.

(10) The stopping tests (with values true or false) on the current manifold are as follows, provided that the solution is feasible; these tests are based, although not identical, in the tests described in /21/.

t1: $\|\bar{h}\|_2 \leq \varepsilon_3 \vee \|\bar{d}\|_2 / (\|1+\bar{X}+\bar{\alpha}\bar{d}\|_2) \leq \varepsilon_4 \vee \bar{Q} = \emptyset$

t2: $|F(\bar{X}+\bar{\alpha}\bar{d})-F(\bar{X})|/|1+F(\bar{X})+\bar{\alpha}\bar{d})| \leq \varepsilon_5$ in the last τ_3 consecutive iterations

t3: $\|\bar{h}\|_2 \leq \varepsilon_6$

t4: $\|\bar{h}\|_\infty \leq \varepsilon_7 T \wedge \tau_4 (|\bar{\lambda}_{B_j} - \bar{\lambda}_{BS_j}|) \leq T$ where

t5: $T > \varepsilon_8$

where $T = \min \{|\bar{\lambda}_{B_j}|, |\bar{\lambda}_{BS_j}|\}$ $j \in \bar{V} \cup \bar{V}_s$

t6: There is not any nonbasic variable with zero or near-to-zero Lagrange multipliers estimate whose tendency is favorable.

An 'optimal' solution is assumed to be found in the current manifold if t1; the current solution is 'quasi-optimal' if $\neg t1 \wedge (t2 \vee t3)$.

When the solution on the current manifold is 'quasi-optimal', both methods use the anti-zigzegging strategy whose main features are as follows: Let us assume that the set D of candidate nonbasic variables (included by the variables with favorable Lagrange multipliers estimates) is already built; note that, in the experiment, the RQN method uses both estimations $\bar{\lambda}_{BS_j}$ and $\bar{\lambda}_{B_j}$ such that they must agree in sign and t4∧t5; the PRNT only uses $\bar{\lambda}_{B_j}$ for which t4∧t5 must be satisfied.A nonbasic variable will not be included in set D if it is an unsafe variable. The set of unsafe variables is included by those nonbasic variables that were made basic-superbasic and, again, become nonbasic in the process of obtaining the next 'optimal' solution. After the de-activating process (even if D=∅), tolerance ε_6 is reset to $\varepsilon_6 = \min\{\varepsilon_3, \varepsilon_6(1-\varepsilon_9)\}$; the unsafe set is declared empty if $\varepsilon_6 = \varepsilon_3$.

When the solution on the current manifold is 'optimal', the unsafe set is declared empty and set D is built with the nonbasic variables with favorable Lagrange multipliers (note that, in this case, $\bar{\lambda}_{B_j} = \bar{\lambda}_{BS_j}$ $j \in \bar{V} \cup \bar{V}_s$) and t5. If D=∅ and t6 /12/, stop since the optimal solution of the problem is found; if ¬t6 then the related nonbasic variable is de-activated. Tolerance ε_6 is reset to $\varepsilon_6 = \varepsilon_{10} \|\bar{\lambda}_B\|_2$, where $\bar{\lambda}_B$ is the vector of basic estimations of the Lagrange multipliers related to set D.

The values that we have used for obtaining the results shown in tables 2 and 3 are as follows: $\varepsilon_3 = \varepsilon_4 = \varepsilon_5 = \varepsilon_8 =$ 10E-04, initial $\varepsilon_6 = \varepsilon_{10} \|h^{(0)}\|_2$, $\varepsilon_7 = 0.9$, $\varepsilon_9 = 0.3$, $\varepsilon_{10} = 0.2$, $\tau_3 = 3$ and $\tau_4 = 2$

(11) Stopping rule I is used for 'solving' the Newton equation in the PRTN method; stopping rule II has not yet been implemented. Tolerance $\bar{\eta}_I$ is obtained by using formula (3.5), where $\bar{g}$ is substituted by $\bar{h}$, $\bar{\eta}_0 = 1/k'$, $\bar{\gamma} = 1$, $\tau_1 = 3\bar{n}_S$ and $\varepsilon_1 = \varepsilon_M^{1/2}$. The tolerance to prevent unstabilities when updating the diagonal BFGS matrix are such that $\varepsilon_2 =$ 10E-04 and $\kappa_m = 1/(100\bar{n}_S^{1/2}\varepsilon_M)$, where $\varepsilon_M =$ 10E-15. Vector $\bar{\delta}^{(i)} \equiv \bar{G}\delta^{(i)}$ is obtained at each minor iteration by using algorithm A5 (see sec. 5.5), where $\sigma = \varepsilon_M^{1/2} / \|\delta^{(i)}\|_2$; the basic-superbasic gradient is analytically evaluated.

(12) The approximate linesearch used in this experiment is the Gill-Murray linesearch (routine GETPC, version 1.981) used in /21/; $\eta = 0.9$ and $\mu =$ 10E-04. An alternative linesearch (whose results are not shown) strongly increase the number of iterations (and, then, gradient evaluations) required to solve Problem II.

The experiment was made in an IBM 370/158 with 3MB of real storage and using VM/CMS, 3MB of virtual storage, the algorithmic tools of MPSX/370 VIL6 and compiler PL/I OPT(2).

The results for Problem I are shown in table 2; both methods are used. Together with the single de-activating strategy, a multiple strategy was also tested up 5 and 20 variables to be de-activated at a time. Strate-

TABLE 1

PROBLEM DIMENSIONS

Problem	m	|E|	n	Variables partition (%)			Density (%)		n_t^*	n_s^*	n_r^*
				Pure L	Non-Pure L	Nonlinear	A	G^*			
I	32	19	88	20	58	22	10	45	26	47	15
II	583	241	1479	24	36	40	1	20	502	459	518

TABLE 2

RESULTS FOR PROBLEM I

Method	De-activating strategy	# Major iters.	# Accumulative minor iters.	# Evaluations of F(X),g(X) in major iters.
RQN	single	132	(132)	310
PRTN	single	86	238	216
RQN	5 vars.	103	(103)	274
PRTN	20 vars.	90	204	263

TABLE 3

RESULTS FOR PROBLEM II

Precondi-tioning	De-activating strategy	# Major iters.	# Accumulative minor iters.	# Evaluations of F(X),g(X) at major iters.
No	single	183	10968	515
	$\tau_2 = 0.01$	212	11460	621
	$\tau_2 = 0.05$	153	10314	402
Diagonal BFGS	single	190	8834	486
	$\tau_2 = 0.01$	205	9372	541
	$\tau_2 = 0.05$	141	7334	384

gies (1)-(12) are used in both methods whenever they are applicable. The PRTN method requires as much as twice the number of gradient evaluations required by the RQN method; but, it does not require to save, nor to update the reduced Hessian matrix. Certainly, more experimentation is needed before any conclusion may be drawn; but, it seems that for small-scale problems, it could be a good strategy to obtain vector $q^{(i)} \equiv \bar{Z}^t \bar{G} \bar{Z} \delta_S^{(i)}$ if A is sparse and G is sparse and available, instead of approximating $q^{(i)}$ by differencing.

The results for Problem II are shown in table 3; it is a super-scale, badly scaled problem and we cannot afford to use the amount of storage required by the reduced Hessian update in the RQN method. So, the experiment was only made with the PRTN method. Several values on the multiple de-activating tolerance have been tested; τ_2=0.01 and τ_2=0.05 gave the best results. The average number of minor iterations per major iteration is here much greater than for Problem I, mainly in the vecinity of the optimal solution on the manifold. The diagonal BFGS preconditioning does not strongly reduce the number of minor iterations but, since its updating does not require large amount of storage, nor it is time consuming, it is interesting to consider it as a default strategy.

Although some results are erratic, we may conclude, based on tables 2 and 3, that deactivating more than one variable at a time generally reduces the number of major iterations.

9. CONCLUSION.

The application of the Truncated Newton methodology to linearly constrained nonlinear programming (LCNP) has been described; it is used for obtaining the superbasic stepdirection, say $\bar{d}_S$ in the frame of a variable-reduction method for LCNP. A procedure for dealing with the reduced Hessian matrix is described; it appears that it is very stable and the required amount of storage is quite acceptable for super-scale problems (i.e., the cardinality, say $\bar{n}_S$ of the superbasic set of variables is between 300 and 600).The reduced gradient is scaled at each major iteration and a 'truncated' strategy is used so that the solving of the Newton equation for obtaining $\bar{d}_S$ needs a smaller number of iterations. The preconditioning is the diagonal BFGS approximation of the reduced Hessian the specialization of the general methodology for updating a product of matrices when one of them is modified (in this case, due to a basic-superbasic exchange or a reduction of the cardinality of nonbasic sets of structural and slack variables) is also fast and stable.

For small-scale problems ($\bar{n}_S$<300), the Quasi-Newton update has the best performance in our limited computational experience. But for super-scale problems, the Quasi-Newton approach seems not to be practical; in this case, the Truncated-Newton approach performs quite satisfactory; the diagonal BFGS used as a preconditioning matrix reduces the number of iterations and its using and updatings require a meaningless amount of storage and time-consumming. The performance of the Truncated-Newton approach could be degradated for non-sparse Hessian and constraints matrices.

The multiple de-activating strategy has generally better performance than the single one, provided that the maximum fraction, say τ_2 of nonbasic variables to be de-activated is not too-large; τ_2=0.05 seems reasonable.

A computational comparison between the Preconditioned Reduced Truncated Newton (PRTN) approach and the Preconditioned 2-steps BFGS with accumulated step (P2BFGSA) /14/ is required for super-scale problems. In the sequel of this paper we are planning to report computational results.

10. ACKNOWLEDGEMENT.

I wish to thank Ron Dembo and Michael Saunders for various guiding comments and suggestions that have strongly improved the implementation of the algorithm. In particular my thanks go to Phil Gill and Walter Murray for providing their linesearch routine.

11. REFERENCES.

/1/ R. BENVENISTE, "A quadratic programming algorithm using conjugate directions", - Mathematical Programming 16 (1978) 63-80.

/2/ A.G. BUCKLEY, "A combined conjugate gradient Quasi-Newton minimization algorithm" Mathematical Programming 15 (1978) ------ 200-210.

/3/ R.S. DEMBO,"Progress in large-scale non-- linear optimization", XI Intern. Mathematical Programming Simp., Bonn, 1982.

/4/ R.S. DEMBO and T. STEIHAUG, "Truncated--- Newton algorithms for large-scale unconstrained optimization", School of Organization and Management working paper", Yale University, Illinois, 1980.

/5/ R.S. DEMBO, S.C. EISENSTAT and T. STEI--- HAUG, "Inexact Newton methods", SIAM J. - of Numerical Analysis 19 (1982) 400-418.

/6/ J.E. DENNIS and J. MORE, "Quasi-Newton -- methods, motivation and theory", SIAM Review 19 (1977) 46-89.

/7/ L.F. ESCUDERO, "A projected Lagrangian -- method for nonlinear programming," IBM -- Scientific Center report G320-3407, Palo Alto (California). 1980.

/8/ L.F. ESCUDERO, "An implementation of the QR-Factorization on solving overdetermi-- ned systems of linear equations," ------- Questiio 4 (1980) 89-94. See also"QR-factorization and its updatings", IBM Scientific Center report SCR-01.82, Madrid --- 1982.

/9/ L.F. ESCUDERO, "An algorithm for large--- scale quadratic programming and its extensions to the linearly constrained case" - IBM Scientific Center report SCR-01.81, - Madrid, 1981.

/10/ L.F. ESCUDERO, "Lagrange multipliers estimates for constrained minimization, --- Questiio 5 (1981) 173-186.

/11/ L.F. ESCUDERO, "Revisión critica del método del gradiente conjugado y extensiones en programación no lineal", Trabajos de Estadística e Investigación Operativa 34 (1983, to appear).

/12/ L.F. ESCUDERO, "Zero or near-to-zero Lagrange multipliers in linearly cons---- trained nonlinear programming," Questiio 6 (1982) 193-204.

/13/ R. FLETCHER, "Unconstrained optimization" (Wiley, London, 1980).

/14/ P.E. GILL and MURRAY, "Conjugate gradient methods for large-scale nonlinear optimization", Systems Optimization Laboratory report SOL 79-15, Stanford University, -- California, 1979.

/15/ P.E. GILL, W. MURRAY and M.H. WRIGHT, -- "Practical Optimization", (Academic Press London, 1981).

/16/ P.E. GILL, W. MURRAY and M.H. WRIGHT, -- "A note on a sufficient decrease for a - non-derivative step-length procedure," Mathematical Programming 23 (1982) 249-352.

/17/ P.E. GILL, W. MURRAY, M. SAUNDERS and -- M.H. WRIGHT, "Truncated Newton methods", XI Intern. Mathematical Programming Symp. Bonn, 1982.

/18/ A.A. GOLDSTEIN and J.F. PRICE ,"An effective algorithm for minimization", Numerische Mathematik 10 (1967) 184-189.

/19/ M.R. HESTENES and E. STIEFEL, "Methods - of conjugate gradients for solving linear systems", J. Res. Nat. Bur. Standards -- Sec. B 48 (1952) 409-436.

/20/ W. KRIBBE, "Nonlinear programming using conjugate directions in reduced dimen--- sions, 6 Symposium uber Operations Re--- search (Athenaum/Hain printers, 1981) -- 113-129.

/21/ B. MURTAGH and M. SAUNDERS, "Large-scale linearly constrained optimization", ---- Mathematical Programming 14 (1978) 41-72

/22/ B. MURTAGH and M. SAUNDERS, "A projected Lagrangian algorithm and its implementa-

tion for sparse nonlinear constraints", Mathematical Programming Study 16 (1982) 84-117.

/23/ L. NAZARETH, "A relationship between -- the BFGS and conjugate gradient algo--- rithms and its implications for new al- gorithms",SIAM J. of Numerical Analysis 16 (1979) 794-800.

/24/ L. NAZARETH and J. NOCEDAL, "Conjugate direction methods with variable storage" Mathematical Programming 23 (1982) 326- 340.

/25/ J. NOCEDAL, "Updating Quasi-Newton ma-- trices with limited storage", Mathema-- tics of Computation 35 (1980) 773-782.

/26/ M.J.D. POWELL, "Algorithms for nonli--- near constraints that use Lagrangian -- functions 14 (1978) 149-160.

/27/ M.J.D. POWELL and PH.L. TOINT, "On the estimation of sparse Hessian matrices", SIAM J. of Numerical Analysis 16 (1979) 1060-1074.

/28/ D.F. SHANNO, "Conjugate gradient ----- methods with inexact searches", Mathema tics of Operations Research 3 (1978) -- 244-256.

/29/ D.F. SHANNO and R.E. MARSTEN, "Conjuga- te gradient methods for linearly cons-- trained nonlinear programming", Mathema tical Programming Study 16 (1982) 149-- 161.

/30/ D.F. SHANNO and P.H. PHUA, "Matrix con- ditioning and nonlinear optimizations", Mathematical Programming 14 (1978) 149- 160.

/31/ P. WOLFE, "Convergence conditions for - ascent methods", SIAM Review 11 (1969) 226-235.